Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

2015-03

# Evaluating the combined UUV efforts in a large-scale mine warfare environment

Thompson, Andrew R.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/45263

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

*Downloaded from NPS Archive: Calhoun*

DUDLEY KNOX LIBRARY

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**EVALUATING THE COMBINED UUV EFFORTS IN A LARGE-SCALE MINE WARFARE ENVIRONMENT**

by

Andrew R. Thompson

March 2015


| | |
|---|---|
| Thesis Advisor: | Susan M. Sanchez |
| Second Reader: | Jeffrey F. Hyink |


**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704–0188* |
|---|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503. | | | |
| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE** March 2015 | **3. REPORT TYPE AND DATES COVERED** Master's Thesis | |
| **4. TITLE AND SUBTITLE** EVALUATING THE COMBINED UUV EFFORTS IN A LARGE-SCALE MINE WARFARE ENVIRONMENT | | | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Andrew R. Thompson | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943–5000 | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | | | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number __N/A__. | | | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release; distribution is unlimited | | | **12b. DISTRIBUTION CODE** |


**13. ABSTRACT (maximum 200 words)**

The current surface mine countermeasures (MCM) fleet is aging, yet there are no viable systems to replace it. The U.S. Navy requires an improved minehunting platform, and unmanned underwater vehicles (UUVs) can meet that need. In order to attain enough UUVs and operators to make these missions successful, the United States must rely on the participation of allies to provide these assets. This study assesses the key decision factors in mine clearance operations using UUVs of differing capabilities. It uses a discrete-event simulation to model the performance of UUVs in a large-scale MCM operation. Data is generated using a state-of-the-art design of experiments and analyzed to find the best tasking plan for the scenario. The results show that with proper tasking, UUVs with lesser ability levels can be used appropriately and still produce acceptable levels of mine clearance, usually more quickly than a smaller cadre of highly capable vehicles. This study finds UUV altitude, track spacing, number of passes, and search speed to be decision factors that influence minehunting results, while track spacing, number of passes, search speed, and resupply are influential factors that effect mission completion times.


| **14. SUBJECT TERMS** | | | **15. NUMBER OF PAGES** |
|---|---|---|---|
| Unmanned underwater vehicles, mine warfare, mine countermeasures, simulation, design of experiments, robust design | | | 103 |
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# EVALUATING THE COMBINED UUV EFFORTS IN A LARGE-SCALE MINE WARFARE ENVIRONMENT

Andrew R. Thompson
Lieutenant, United States Navy
B.S., United States Naval Academy, 2007

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN OPERATIONS RESEARCH**

from the

**NAVAL POSTGRADUATE SCHOOL**
**March 2015**

Author: Andrew R. Thompson

Approved by: Susan M. Sanchez
Thesis Advisor

Jeffrey F. Hyink
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The current surface mine countermeasures (MCM) fleet is aging, yet there are no viable systems to replace it. The U.S. Navy requires an improved minehunting platform, and unmanned underwater vehicles (UUVs) can meet that need. In order to attain enough UUVs and operators to make these missions successful, the United States must rely on the participation of allies to provide these assets. This study assesses the key decision factors in mine clearance operations using UUVs of differing capabilities. It uses a discrete-event simulation to model the performance of UUVs in a large-scale MCM operation. Data is generated using a state-of-the-art design of experiments and analyzed to find the best tasking plan for the scenario. The results show that with proper tasking, UUVs with lesser ability levels can be used appropriately and still produce acceptable levels of mine clearance, usually more quickly than a smaller cadre of highly capable vehicles. This study finds UUV altitude, track spacing, number of passes, and search speed to be decision factors that influence minehunting results, while track spacing, number of passes, search speed, and resupply are influential factors that effect mission completion times.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AMCM | Air mine countermeasures |
| DOE | Design of experiments |
| DTE | Detect to engage |
| EOD | Explosive Ordinance Disposal |
| FY | Fiscal year |
| IMCMEX | International Mine Countermeasure Exercise |
| LCS | Littoral Combat Ship |
| MCM | Mine Countermeasures |
| MILCO | Mine-like Contact |
| MTA | Mine Threat Area |
| NAVCENT | U.S. Naval Forces Central Command |
| NOMBO | Non-mine mine-like bottom object |
| PMA | Post-mission analysis |
| RMMV | Remote Multi-Mission Vehicle |
| RMS | Remote Minehunting System |
| SAS | Synthetic aperture sonar |
| SLOC | Sea lines of communication |
| SMCM | Surface mine countermeasures |
| UMCM | Underwater mine countermeasures |
| UNCLOS | United Nations Convention of the Law of the Sea |
| UUV | Unmanned underwater vehicle |

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

Mines are the single most effective and cost efficient weapons known to naval warfare. They are vicious tools that can block all sea trade and prevent security and supplies from entering a particular region. Minefields can take weeks, months, or even years to clear, with no real certainty of completion. To underestimate the capability of today's mines can prove fatal.

The current surface mine countermeasures (MCM) fleet is aging, yet there are no viable systems to replace it. The U.S. Navy requires an improved minehunting platform and unmanned underwater vehicles (UUVs) can meet that need. The U.S. needs support from the international community to provide MCM Commanders with enough UUVs and operators to make these missions successful. When countries form new partnerships with the U.S. it remains difficult to assess their abilities to execute MCM. This unfamiliarity makes tasking their UUVs challenging, because their skill level is unknown.

This study show evidence that it is possible to use an efficient planning design to incorporate all international UUV assets, attain desired clearance levels, and finish in a reasonable timeframe. A discrete-event simulation is used to model the execution of an MCM scenario.

The model is written in Python programing language and the flowchart in Figure 1 shows the logic and sequence of events. The first part of the simulation is constructing the Q-route, and setting the simulated mines. UUVs then drive search tracks inside the Q-route until the coverage is complete. Following the search, a post-mission analysis is conducted to detect and classify all mines as mine-like contacts (MILCOs) and all non-mine mine-like objects (NOMBOs). Next, all bottom contacts are reacquired and identified using a star pattern UUV search. Once all mines are identified, the explosive ordinance disposal (EOD) dive platoons neutralize them.

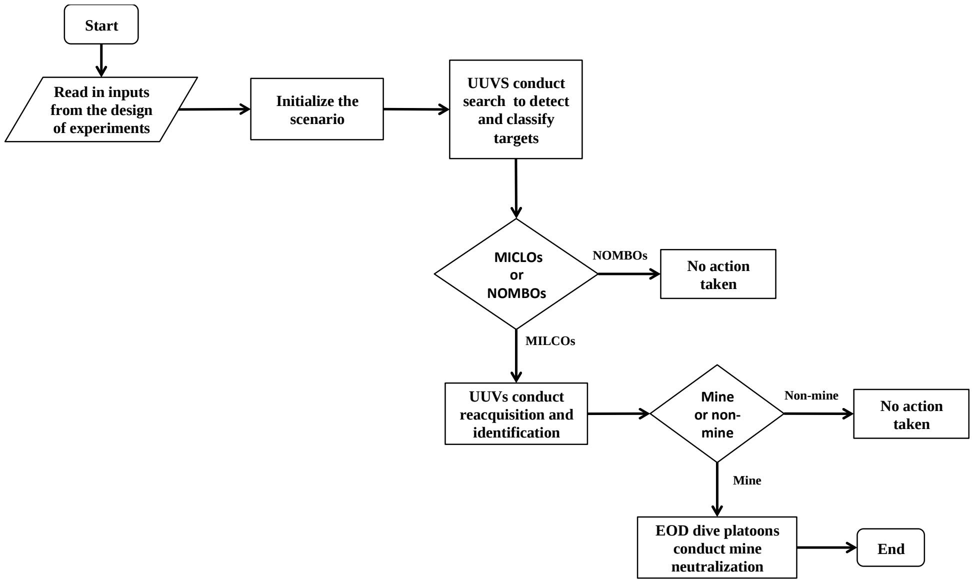



Figure 1 Flowchart showing the for the MCM UUV model.

Data is generated using a state-of-the-art design of experiments and then analyzed to find the best tasking plan for the scenario. The results found that UUV altitude, track spacing, number of passes per track, along with search speed, all influence the post-mission analyst's ability to detect bottom objects. The decision factors that effect mission completion times are: search speeds, number of passes, and track spacing. These factor settings were manipulated using a robust design so that the proportion of undetected objects and MCM mission completion times were both minimized.

Another simulation was conducted to focus solely on the performance of American MCM operations. In this scenario only UUVs with outstanding capabilities are used, but in far fewer numbers than the coalition simulation.

The comparison of results shows that the coalition force outperforms in both detection effort and overall mission completion times. These results provide evidence that a coalition UUV force is more than qualified, even with a mix of experience levels and capabilities.

# ACKNOWLEDGMENTS

This thesis would not have been possible without the support of many people. In particular, I would like to thank Dr. Susan Sanchez, for being an extraordinary advisor through countless meetings and revisions. I would also like to thank CAPT Jeffrey Hyink, for being an invaluable mentor and teacher that I could always count on for true guidance. Thank you to Chris Avedissian, for being the greatest friend and surf buddy, COL Fred Lough and Mary Mullin, for being loving and support parents-in-law, and Linda and Harry Thompson, for being the most devoted and wonderful parents ever. Finally, I'd like to thank my family, Jessie, Mary, and Cora, the loves of my life, for inspiring me to be a better person and for making every day the best adventure.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Freedom of navigation has been a pillar of global commerce throughout history. To this day, the majority of goods and crude oil flow from nation to nation by sea. Sea lines of communication (SLOC) provide merchant traffic with routes through open ocean, as well as narrow straights and territorial waters. These routes are protected under the United Nations Convention of the Law of the Sea of 1982 (UNCLOS). The UNCLOS states that all nations have right of innocent and transit passage through international waters and exclusive economic zones as long as the transiting ship poses no physical or economic threat to the coastal nation (United Nations, 2015, p. 31). The United States Navy is dedicated to upholding these laws by maintaining freedom of the seas for all nations in good standing with the United Nations. This commitment helps ensure a more stable global market, as well as provide access of United States military forces to coastal nations, both friendly and hostile.

## A. BACKGROUND

Naval mines pose one of the greatest threats to free navigation of the seas. A single mine could potentially close entire ports, straits, anchorages, channels, or any other bodies of water. They are small and extremely difficult to find, making it is nearly impossible to determine how many mines are in a designated area. Therefore, the mined waterways must undergo extensive de-mining operations before the minefield is clear and normal traffic can resume. For this reason, it is sometimes advantageous for a nation to falsely declare an area to be mined, because the slightest uncertainty will force an enemy halt their current mission and deal with the potential mine threat.

Mine countermeasures (MCM) are naval operations devoted to removing sea mines. It is a vital mission area for the U.S. Navy. The U.S. Navy maintains a strong MCM force, with surface ships, helicopters, and underwater systems, all structured to locate and destroy mines. New technologies allow for safer and more reliable systems to replace the older systems. The Littoral Combat Ship (LCS) MCM Mission Module is one

of these new systems. However, the LCS MCM Mission Module program is significantly behind schedule. A potential answer is temporarily replacing Avenger-Class ships with unmanned underwater vehicles (UUVs) until the LCS MCM Mission Module is ready. This solution is viable, but needs large numbers of UUVs and operators from partnering nations around the world.

### 1. Threats

While mines are designed to inflict significant damage to ships, even strategically placed mines are not necessary intended to strike any vessel in particular. The threat alone attains the desired outcome. They prevent the flow of marine traffic. They can be used to prevent enemy forces from entering a coastal nation’s waters. Naval mines can also be used offensively by blocking another coastal nation from entering and leaving their own port. This tactic is not a permanent solution, but mining can give the mining nation valuable time to prepare other military forces (U.S. Joint Chiefs of Staff, 2011, p. II-7).

Naval mines exist on a broad spectrum of sophistication. The most simple naval mine is the contact mine. These mines are moored to the ocean floor and are suspended in the water. They detonate on impact with a ship. There is no internal logic or sensor. Due to their simplicity, they are the cheapest, but also the easiest to remove. Influence mines are more complex than contact mines. They have different types of sensors that can target specific types of ships. Sensors can react to vibration, magnetism, and pressure. They generally sit on the ocean floor and wait for ships to pass over them. Every ship class is unique and each engineering plant creates different mechanical vibrations. The different amount of metal generates a different magnetic signature. Each ship’s hull produces a different wake. The influence mines can be tuned to target specific ship classes.

Minesweeping is a mine countermeasure operation that aims to destroy mines without knowing exactly where the mines are located. The de-mining ships or aircraft drag equipment that emits the specific magnetic signature, vibration, or pressure in order to mimic military ships. The goal is to trigger the mines along a certain path so that the

real ships can pass through a cleared area. Minesweeping is quick and easy, but easy to dodge. Smarter mines are equipped with ship-counters, which is a built-in function that counts the number of ships that pass above. It allows the minelayer to deceive the demining force by not detonating on the first few passes. It will detonate only after a pre-determined number of passes. This mine counter-countermeasure creates ambiguity for the de-mining nation, and the resulting uncertainty prevents the assurance of safe passage. Therefore, minesweeping is a very effective last resort countermeasure, but is not the primary means to clear a minefield. The only way to clear mined waters is to find all mines in a path and destroy them (U.S. Joint Chiefs of Staff, 2011, p. IV-8).

### 2. Defense

Finding and destroying mines is the process known as mine hunting. Unlike minesweeping, mine hunting uses sensors to search an entire region of water. Bottom objects are detected. If during the classification process, an object's sonar echo is mine-like in shape, but assumed not to be a mine, the object is then categorized as a non-mine mine-like bottom object (NOMBO). If the object shape is mine-like, then the object is classified as a mine-like contact (MILCO). That contact is then carefully inspected, either with camera, lasers, side-scan sonar, or explosive ordinance disposal (EOD) divers. If the visual inspection identifies a mine, then it can be neutralized, usually by a controlled detonation.

Mine hunting is a very tedious procedure. It takes a lot of time and effort. Even if the search area is completely covered, there is still a chance that some mines will not be found. To minimize the chance of missing mines, a higher level of effort is required to raise the percentage of clearance. In order for the mine threat to be considered cleared, there must be a high confidence that all mines are removed. Sometimes a desired clearance level is not possible due to environmental conditions. This is usually determined by the amount of bottom clutter and type of seabed. A heavy bottom clutter and a rocky bottom lower detection and classification rates. A hard seafloor reflects more sound than a soft bottom; this echo diminishes the SONAR clarity and makes detection

challenging. A highly cluttered bottom makes classification difficult and leads to incorrectly classifying non-mines as MILCOs and mines as NOMBOs. Conversely, fewer bottom objects and a soft bottom make detections and classification easier.

The MCM force is composed of three subgroups: Air MCM (AMCM), Surface MCM (SMCM), and Underwater MCM (UMCM). AMCM uses MH-53E helicopters to conduct minesweeping and minehunting operations. UMCM uses EOD divers, marine mammals, and unmanned underwater vehicles (UUV) to conduct clearance and neutralization operations. SMCM is composed of the Avenger-Class ships. SMCM provides the most sustainable platform to conduct all forms of MCM. The Avenger-Class ships are also equipped to deploy some UMCM assets. Additionally, the ships are equipped with the manpower to remain on task during extended operations, day and night (U.S. Joint Chiefs of Staff, 2011, pp. E-2 - E-4).

The Avenger-Class ships conduct minehunting operations by first entering a minefield with high-definition sonars pointed in the forward direction. This approach allows sonar operators to scan the water directly in front of the ship before advancing. This keeps the ship safe from potential mine detonations. If a mine is detected, it can then be identified and neutralized. When directed to identify and neutralize a mine-like contact, the ships deploy the mine neutralization system, either the AN/SLQ-48 Mine Neutralization Vehicle or the SeaFox. Both systems are remotely operated and guided to the mine-like contact using their sonar reflections. Each system has a built in camera used to visually inspect the contact. The object can only be identified as a non-mine after visual inspection. If it is positively identified as a mine, or uncertainty exists as to whether it is or is not a mine, then the AN/SLQ-48 will place an explosive package next to the mine. The vehicle is then recovered and the package and mine are detonated. The difference between the AN/SLQ-48 and the SeaFox system is that the SeaFox will detonate itself along with the mine. It is both a sensor and a munition. Both of these systems are very effective in conducting mine clearance (Federation of American Scientists, 1999).

The future surface MCM platform is the MCM mission module for the Littoral Combat Ship. This system conducts the full detect-to-engage (DTE) sequence for bottom influence mines. Included in the MCM module is the Remote Minehunting System (RMS). It is composed of the Remote Multi-Mission Vehicle (RMMV) and the AQS-20 sonar. The RMS navigates to the area of interest and drives tracks while the ship remains outside of the minefield. Once its mission is complete, the RMS is recovered and its data are downloaded for post-mission playback. This playback is referred to as post-mission analysis (PMA). An operator inspects the sonar data, looking for mine-like contacts. All contacts are reported and neutralized later by the embarked airborne assets.

The MCM mission module is currently scheduled to be available in 2015; however, most recent testing suggests otherwise. Early in fiscal year (FY) 2015, several key issues were identified with the RMS. The mean time between operational failures is 34.6 hours, rendering the system officially unreliable. The failure is described as erratic and undesired movements. Additionally, the Independence Class LCS is having problems recovering the RMMV. The RMMV is not the only system failing to meeting performance benchmarks. The AQS-20B is scheduled to replace the AQS-20A in 2015. This upgrade includes replacing the side-scan sonar with forward-looking sonar and synthetic aperture sonar (SAS). These improvements increase resolution and diminish altitude distortion caused by irregular sonar echoes. This program is also having issues, and may not be completed in 2015. Until the RMMV and AQS-20B issues are resolved, the RMS operational test cannot be completed onboard LCS (Seligman, 2015).

These poor results were followed by Congress' decision to zero out the RMS budget for FY-2015. There is speculation that this decision may be grounds to abandon the current RMS program and start fresh. With all of the issues plaguing the MCM mission module, is not likely that an MCM capable LCS will be available in the foreseeable future. Therefore, the Avenger-Class ships will remain the main MCM platform (Seligman, 2014).

The Avenger-Class ships are approaching the end of their service lives. Due to the urgent operational needs of combatant commanders, the decommissioning date will likely

be postponed indefinitely, and legacy systems will continue to be funded and upgraded as necessary (Allen, 2013). Though this ad hoc approach is temporary, it will become increasingly costly as the ships and systems age. Eventually, the ships and equipment will degrade beyond the capability of feasible repair, at which point they will be forced to be retired. The big question with this plan is if it can be sustained long enough for a sufficient number of LCS ships to relieve the MCMs.

### 3. Potential Resolution

The inevitable retirement of the Avenger-Class fleet and the slow rollout of LCS ships create a potential capability gap. Combatant commanders must rely on the UMCM force to fill this gap. The EOD teams are able to conduct minehunting operations using the MK18 Mod 1 REMUS 100 and the MK18 Mod 2 REMUS 600 UUVs. They are both equipped with side-scan sonar, which records imagery of the ocean bottoms. The MK18 Mod 2 is shown on a rigid hull inflatable boat in Figure 1. After a mission, the UUV sonar data is downloaded and analyzed to find mines. While very effective, this process is very time intensive and is relatively manpower heavy. EOD mobile units are not able to scale up their efforts to match the capability of the SMCMs. A solution to this manpower shortage is incorporating UUVs and EOD dive teams from other nations.

International Mine Countermeasures Exercises (IMCMEX) in 2012 and 2013 were large-scale exercises hosted by U.S. Naval Forces Central Command (NAVCENT). The two exercises showed growing interest in supporting combined MCM operations. IMCMEX 2012 observed over 30 participating nations, and IMCMEX 2013 had over 40 participants. Not only did the exercises show combined support, they also highlighted several areas of difficulty. In 2012, one of the problems was the difficulty in merging UUV information from newly participating countries. Following a UUV mission, the operators would conduct an analysis of the sonar data and report their findings. Since the United States and the United Kingdom normally operate together in Fifth Fleet, their findings were automatically reconciled. Other participant abilities were less familiar. There was no method to verify the quality of their results, and therefore, the MCM

Commander was not able to use their results. This problem is one of the most substantial out of all lessons learned. If data collected by other nations' UUVs cannot be used, then many of these countries will be discouraged from participating in future operations (Naval Mine and Anti-Submarine Warfare Command, 2013).

Figure 1. Sailors deploying the MK 18 Mod 2 Swordfish UUV in FIFTH FLEET during IMCMEX 2014 (from Midnight, 2014).

## B. SCOPE

This study identifies operational factors that have the greatest influence on mine clearance levels and completion times. It investigates the advantages and disadvantages of incorporating less capable UUVs into large-scale MCM operations. It also develops a baseline procedure for tasking UUVs with different abilities.

This study uses a simulation model and a state-of-the-art design of experiments to determine the best approach to conduct combined mine clearance operations using UUVs

in Fifth Fleet. Chapter II is a literature review that identifies current research and development of UUVs. It also illustrates the importance of this research topic. Chapter III discusses the methodology. It examines the model structure, variables, constraints, limitations, and assumptions, and provides details about the design of experiments used to explore the simulation model. Chapter IV explains the analysis process using a robust design. Chapter V is the conclusion, where recommendations and future work are described.

# II. LITERATURE REVIEW

Mine clearance operations are extremely dangerous. Humans are often required to enter a minefield in order to render mines safe. Fortunately, as more advanced unmanned systems come online, the need to have humans in mined waters decreases. Current mine countermeasures research is dominated by UUV design and autonomy. On-the-spot autonomous detection is one area of interest. Synthetic aperture sonar (SAS) is becoming the new standard for UUV sonars. It is replacing the side-scan sonar. This upgrade allows UUVs to collect imagery that has significantly higher resolution, which is better suited for target recognition algorithms. For more information, see Sternlicht, Fernandez, and Marston (2013).

## A. AUTONOMOUS DETECTION AND CLASSIFICATION

The ocean floor is highly varied environment. Without traveling too far, the bottom can change from sand to rock, smooth to rough, shallow to deep, or even clean to cluttered. These changes present a significant challenge for autonomous detection software. Bottom conditions influence how instruments are calibrated. Current recognition software requires sensors to be carefully tuned prior to conducing searches. If bottom conditions change, then the sensors and software will not be as effective. These algorithms are so sensitive that small sand ripples can throw off detection and classification rates. They need a uniform surface in order to be successful. This constraint is not realistic, which is why new software is designed to adapt to changing environments. Algorithms are being developed that focus on objects' shapes rather than their contrast to the surroundings, but this new software is limited to what it can see. The side-scan sonars are extremely sensitive to altitude change and speed changes. Tiny speed changes or altitude shifts will cause blurring that will throw off these algorithms. The high-resolution picture of the SAS is far more resistant to speed changes and altitude shifts, and can map a clearer picture of the bottom and bottom objects. The SAS produces

sharp shadows behind the bottom objects, which are then used to determine if the object is a mine or not (Leier, Fandos, & Zoubir, 2015, p. 71).

Previous autonomous algorithms were designed to process images with fewer pixels. Image processing slows down after integrating the high resolution SAS imagery with the old detection algorithms. New algorithms are being developed and refined to speed image processing for near-real-time detection and classification (Maurelli, Patron, & Cartwright, 2011).

Ocean currents sometimes push sand on top of mines. The sand acts as a shield, making the mine invisible to sonar. The Marine Mammal System is the only MCM asset in the US Navy that can detect buried mines. These mammals are not always available, so there is demand for UUVs to also provide this capability. Current research is studying the ability to detect buried mines using an array of magnetic, acoustic, and electro-optic sensors. The aim is to search an area with a UUV and collect data with all three of these sensors. The fusion of these data streams then feeds into the autonomous detection software. As is the case with other autonomous research, results are environmentally dependent. Poor water visibility decreases performance (Sulzberger et al., 2009).

## B. REACQUISITION AND NEUTRALIZATION

Sonar imagery from side-scan sonars and SAS provides sufficient detail to identify contacts without having to make a visual confirmation. The UUVs drive star-like patterns above contacts in order to capture multiple aspects needed to make the identification. This technique is being adapted for autonomous identification. UUVs will calculate their own path and conduct a star pattern for each contact. This approach can be redundant in densely littered minefields, where multiple star patterns overlap. One current research goal is to develop algorithms for smarter path planning. The Multiple Aspect Coverage pattern is one of these algorithms, shown in Figure 2. The performance of this algorithm was analyzed using a Monte Carlo simulation and compared to the performance of standard pattern. Results show that the Multiple Aspect Coverage pattern

requires 29% less travel distance than standard patterns for reacquiring and identifying MILCOs as mines in densely packed clusters of mines (Bays, 2014).

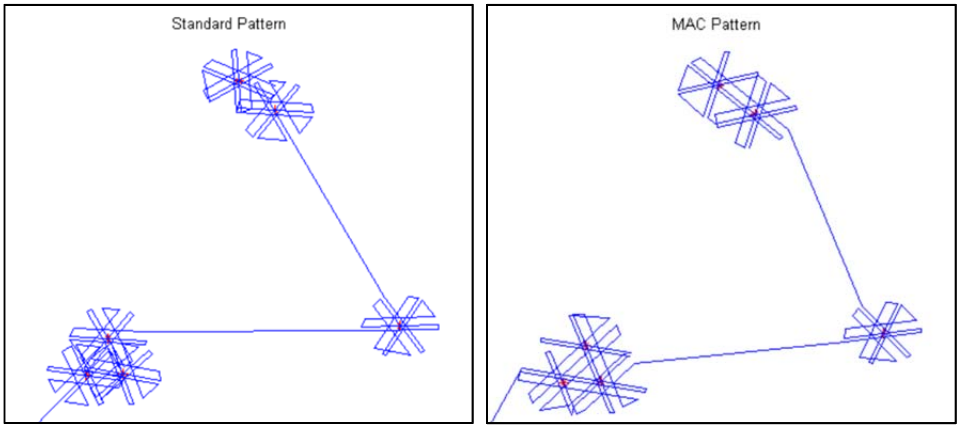



Figure 2. Standard pattern (left) and Multiple Aspect Coverage pattern (right) (from Bays, 2014).

## C. MINE NEUTRALIZATION

Detection and classification are not the only areas of research activity. Neutralization is another area of focus. Current doctrine directs MCM units to detonate mines, because defusing mines is too risky. An inadvertent mine detonation could kill operators. There is no risk to human life if UUVs are conducting the neutralization. Therefore, UUVs can neutralize mines' threats by deactivating them instead of blowing them up. This functionality requires mechanical arms to interact with the mines. Kemp et al. (2011) conducted research to find the optimal position to mount these arms to improve functionality and reduce drag. This study also identifies the thrust requirements to maintain overall stability and position.

UUVs capable of autonomous detection, classification, identification, and neutralization nearly complete the detect-to-engage sequence. The final piece to a fully autonomous system is communications. Since the DTE sequence cannot be completed by just one UUV, an updated list of targets would need to be maintained at a command

center. Search UUVs would upload their results to the command center. The target data would then be forwarded to other UUVs to be neutralized. Techniques for transferring data are being developed using underwater lasers and electromagnetic wave propagation. Initial studies show that this method of data exchange is possible and is also very fast, but further research is needed to account for long distance missions and poor ocean visibility (Song & Chu, 2014).

## D. REFINE CURRENT TACTICS

Research suggests that UUVs will soon replace other MCM assets. Autonomous detection will improve clearance times and reduce the number of human operators. But until these systems are available, UUVs will be delivered and recovered by human operators. Humans will parse sonar data, searching for mines, and EOD teams will manually neutralize mines. This process is manpower heavy and will require tremendous support from the international community. But in order to task these other partners, the MCM Commander needs to know their capabilities. Simulating the performance of UUVs in a mine clearing scenario will provide insight on how best to task UUVs.

Several others have addressed various aspects of the mine clearing problem. Allen (2004) and Allen, Buss, and Sanchez (2004) looked at search and detection—one of the primary roles of UUVs in the U.S. Navy. They used a discrete event simulation to analyze a UUV’s ability to detect mines in an environment that causes navigation error. The results suggested that the inaccuracy of the dead reckoning function in the UUV, along with currents that introduce additional navigation error, reduce the post-mission analyst’s ability to detect mines. This study found that hunting along the direction of the current reduces error, and using a transducer to help the UUV navigate underwater, produces better detection results.

Not all searches are done underwater with an unknown number of targets. Search and rescue operations are sprawling search problems on the surface of the ocean. These operations generally involve one target, aerial search assets, and a surface vessel to recover the person. Ashpari (2012) created a spreadsheet model to examine the ability of

aerial vehicles to locate the person, while the surface vessels must race to recover him or her. The model uses the inverse cube law to calculate probabilities of detection for the aerial search. The results find that the most influential factor in successful search and rescue is a fast ship. The second biggest contributor is the number of unmanned aerial vehicles. This research shows that multiple search assets can improve the ability to detect objects.

Sometimes it does not pay to have better detection capabilities. A simulation study by Kim (2002) investigated a ship's ability to safely sail through a minefield using onboard sensors that detect the presence of mines. The results showed that the mine detection systems would give multiple false alarms in a minefield with high bottom clutter. These false alarms forced the ships to change course, which increased their time in the minefield and increased their risk of detonating a mine. These results highlight the importance of classifying bottom objects. By not attempting to distinguish between mines and non-mines, the detection system assumes everything is a mine and recommends a more dangerous path through the minefield.

Team Mine Warfare (MIW), from the Naval Postgraduate School's Systems Engineering department, developed a discrete event model in ExtendSim8 to compare current MCM systems against the future LCS MCM Mission Module (Blandin, et al., 2014). The DTE process for LCS is similar to the DTE for UUVs and EOD dive platoons. The RMS conducts the search and the MH-60S conducts identification and neutralization. This simulation is useful for studying UUV performance, but regrettably, has issues that make it unsuitable for the application in this thesis. The first issue is that they chose to use a cookie cutter sensor, and set the search sensor range equal to the track spacing. This does not reflect reality. Sensors are less effective as distance increases, and narrow track spacing allows for overlapping sensor opportunities and thus better detection. The second issue is an apparent software glitch, which was revealed after running the simulation through a large design of experiments (512 carefully chosen excursions, called "design points"). Inputs with fewer than 15 mines and 30 non-mines produce no output—and this represented one third of the design points. This may not

have affected the results in Team MIW's study because their excursions involved hundreds or thousands of mines, but it does suggest that further V&V on their model is needed before the results are used for making actionable recommendations. Finally, Team MIW's analysis was based on a single replication of each design point. In this study, we are specifically interested in examining the variability associated with the MCM outcomes, in order to better understand the potential risks involved, and seek solutions that are robust to variability in the operating environment.

The more international partners participating in an MCM operation, the more UUVs will be available to conduct searches. With proper tasking, these UUVs may help reduce completion times while maintaining proper clearance levels. In order to provide this tasking, MCM Commanders need guidance on how to employ these international partners. This study aims to provide that guidance.

# III. METHODOLOGY

The simulation developed for this thesis models a real life MCM scenario using UUVs in an open ocean environment. Though EOD Unmanned Systems Platoons are structured to operate in a very shallow water (VSW) zones, the simulation uses their VSW procedures in a large-scale open ocean operation.

## A. THE SCENARIO

In order to create a suitable simulation, it is important to first create a suitable conceptual scenario to use as a baseline. While there is no precedent for executing large-scale MCM operations with only UUVs, the procedure would likely follow a similar method as a smaller operation in VSW. The main difference would be the increased number of UUVs needed to clear the area.

### 1. Planning

The first phase in conducting an MCM operation is planning. The MCM Commander decides on a course of action that best matches the situation. For the purposes of this study, we assume the area being cleared is a generic Q-route, which is a channel 30 nautical miles (NM) long and 0.9 NM wide, that designates the area to be de-mined. Figure 3 shows a picture of the Q-route. The Commander is located outside of the minefield onboard the base ship. We also assume there are a total of 30 UUVs available from the U.S. and several allied countries, and 10 experienced EOD platoons. These numbers are for illustration purposes, but our approach can serve as a template for investigating scenarios involving other numbers and types of UUVs. First, the UUVs are used to search the entire area. The mines identified are later be neutralized by EOD platoons.

The Q-route is broken down into five rows. The first and fifth rows are 0.1 NM wide, the second and fourth rows are 0.2 NMs wide, and the middle row is 0.3 NM wide. Each row is divided into five smaller areas. The total number of individual areas is 30,

with six areas per row. Each UUV is tasked to hunt in one of these areas. The purpose of using areas of different sizes is to give the MCM Commander the option of putting less capable UUVs on the outside of the area, perhaps with lower track spacing, and keeping the more proficient UUVs in the middle. We compare our results to those that would be achievable if the only available UUVs and EOD teams were U.S. Navy assets.

### 2. Search

The platoon delivers each UUV to its designated search area via a rigid hull inflatable boat (RHIB). Once deployed, the UUV conducts a lawnmower search pattern, as shown in Figure 3. Each UUV starts its search from the southwest corner of its region, and initially heads east. Each UUV starts every mission with fully charged batteries. It can then search for a predetermined amount of time before it must be recharged. If the UUV runs out of battery life, it finishes its current track, is recovered by the RHIB, and brought back to the base ship. Sonar data are downloaded and the batteries are recharged. The post-mission analyst reviews the sonar imagery from the mission, looking for mine-like objects. If objects are detected, they are classified as mine-like contacts (MILCOs) or a non-mine mine-like bottom objects (NOMBOs). The MILCO positions are forwarded to the MCM Commander, where they are added to the list of other contacts. If the UUV was not able to complete its search on the previous mission, it will redeploy back to the Q-route to continue where it left off. This process is repeated for all UUVs until the entire area has been searched. If it does complete the search before the battery life expires, it returns to the RHIB. It is recovered and brought back to the ship. The data are again downloaded and post-mission analysis is conducted. The batteries are charged to prepare it for its follow-on mission of reacquisition and identification.

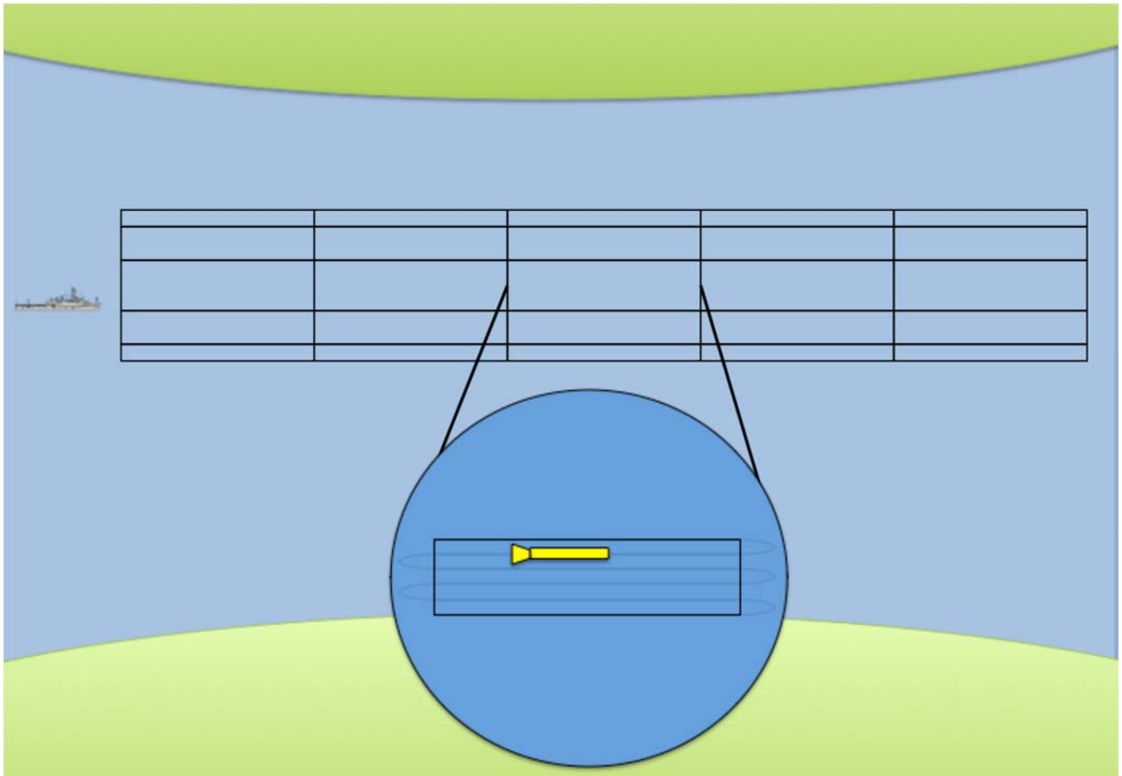


Figure 3. Illustration of the Q-route and the search areas and direction of search tracks.

### 3. Reacquisition and Identification

Once the search phase is complete, the UUVs conduct further investigation to categorize MILCOs as non-mines or mine. The UUV is transported back to the search area where it performs a star pattern search above all MILCOs. It records sonar data and visual video. The dual data streams provide sufficient information to identify the MILCO contact as a mine or a non-mine. This process is called reacquisition and identification.

### 4. Neutralization

Once reacquisition and identification is complete, the EOD platoons are tasked to neutralize all mines. The platoons travel through the minefield via RHIBs to the locations of each mine. They dive on top of the mine, place an explosive neutralizer around the mine, and detonate it. The explosion destroys the mine.

## B. THE SIMULATION

The simulation is a stochastic discrete-event model implemented as a Python program. The logic and structure of the model follows the scenario described earlier in this chapter. The flowchart in Figure 4 shows the sequence of events for the model.

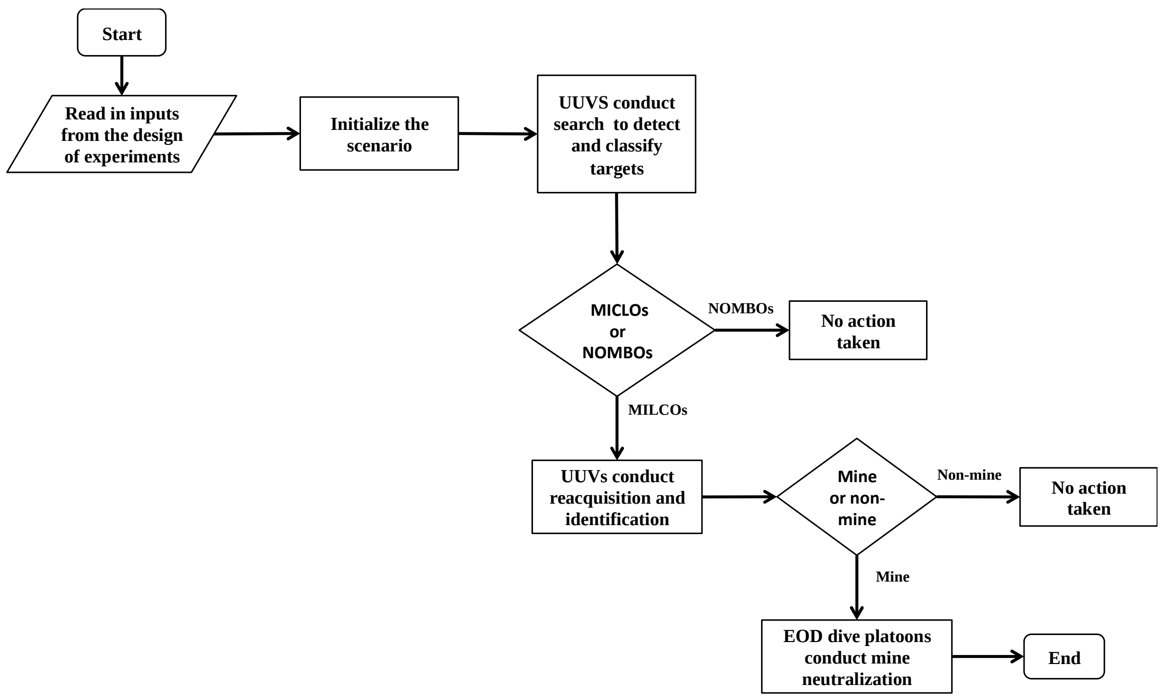



Figure 4. Flowchart showing the broad logic scheme of the simulation model.

### 1. Building the Area

The first part of the simulation is constructing the Q-route. Area objects are created for each UUV. The areas are then stacked together to form a large Q-route area object. The bottom objects are generated and uniformly scattered throughout the Q-route. Some of the objects are mines and some are non-mines. Figure 5 explains the logic for this phase.

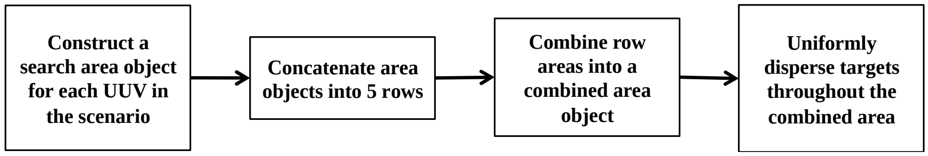



Figure 5. Flowchart showing the logic of the scenario setup.

### 2. Conducting the Search

Thirty UUV objects are created and paired with an area within the Q-route. Every UUV starts the scenario with fully charged batteries. It will drive tracks in its assigned area until the coverage is complete. The detection and classification are done during the post-mission analysis (PMA), after a mission has been completed. The results from the PMA are based on the probabilities of detection and classification are computed for each target, and recalculated for every pass made by the UUV. These probabilities are calculated using the inverse cube law, which is a cumulative detection probability function that calculates the probability of detecting or classifying each target at least once per track (Chung, 2014). The equations for detecting and classifying mines are:

$$P\left(detect\right) = 1 - e^{\frac{-2 \cdot detRate \cdot A \cdot altitude}{searchSpeed\left(altitude^2 + distance^2\right)}} \tag{1}$$

$$P\left(classify\ as\ MILCO \middle| detect\right) = 1 - e^{\frac{-2 \cdot milcoRate \cdot A \cdot altitude}{searchSpeed\left(altitude^2 + distance^2\right)}} \tag{2}$$

$$P\left(classify\ as\ MILCO\right) = P\left(detect\right) \cdot P\left(classify\ as\ MILCO \middle| detect\right). \tag{3}$$

Similarly, the equations for detecting and classifying non-mines are:

$$P\left(detect\right) = 1 - e^{\frac{-2 \cdot detRate \cdot A \cdot altitude}{searchSpeed\left(altitude^2 + distance^2\right)}} \tag{4}$$

$$P\left(classify\ as\ NOMBO \middle| detect\right) = 1 - e^{\frac{-2 \cdot nombosRate \cdot A \cdot altitude}{searchSpeed\left(altitude^2 + distance^2\right)}} \tag{5}$$

$$P\left(classify\ as\ NOMBO\right) = P\left(detect\right) \cdot P\left(classify\ as\ NOMBO \middle| detect\right) \tag{6}$$

where *P(detect)* is the probability of detecting each object at least one time per pass, *P(classify NOMBOs)* is the probability of classifying each non-mine as a NOMBOS at least one time per pass, and *P(classify MILCOs)* is the probability of classifying each mine as a MILCO at least one time per pass. The exponent in these equations describes the detection or classification rates for each target. *A* is the area of the target, *altitude* is the height of the UUV from the ocean bottom, *searchSpeed* is the speed of the UUV, *distance* is closest point of approach (CPA) from the UUV to the target, and *detRate*, *nombosRate*, and *milcoRate* are shape parameters that describe the post-mission analyst's ability to detect and classify mines and non-mines. These shape parameters also scale the detection rate to a size that is applicable to minehunting, as opposed to a vast aerial search and rescue operations. They also allow us to model different classification behaviors. For example, countries less sure of their capabilities might be more inclined to report objects that they are uncertain about as MILCOs regardless of their true type, in order to avoid the risk of missing mines. These shape parameters, with the rest of the inputs, create the probabilities to each bottom object.

The inverse cube law is suitable for modeling detection from UUVs because UUVs search from above while moving forward at a certain velocity. The UUVs look down and outward, searching for objects on the ground where closer targets are easier to detect and classify than farther targets. For example, suppose a UUV has a detection rate of 0.05, an altitude of eight meters, and a search speed of four meters per hour. It will detect a mine that is 22 meters away with a probability of 0.50 and it will detect a mine that is 10 meters away with a probability of 0.90.

After the probabilities are calculated, a uniform random number is generated for each target. The target is detected when the random number is less than the calculated detection probability. Otherwise, the target is undetected. Classifying a target is only possible if it is first detected. Therefore, correct classifications require the random number to be less than the product of the detection and classification probabilities. Otherwise, the target is falsely classified. However, if a target is undetected or falsely classified during the first pass, it can be reassessed on the next pass. A target is not

reassessed if it has already been correctly classified. Figure 6 shows the flowchart for this phase.

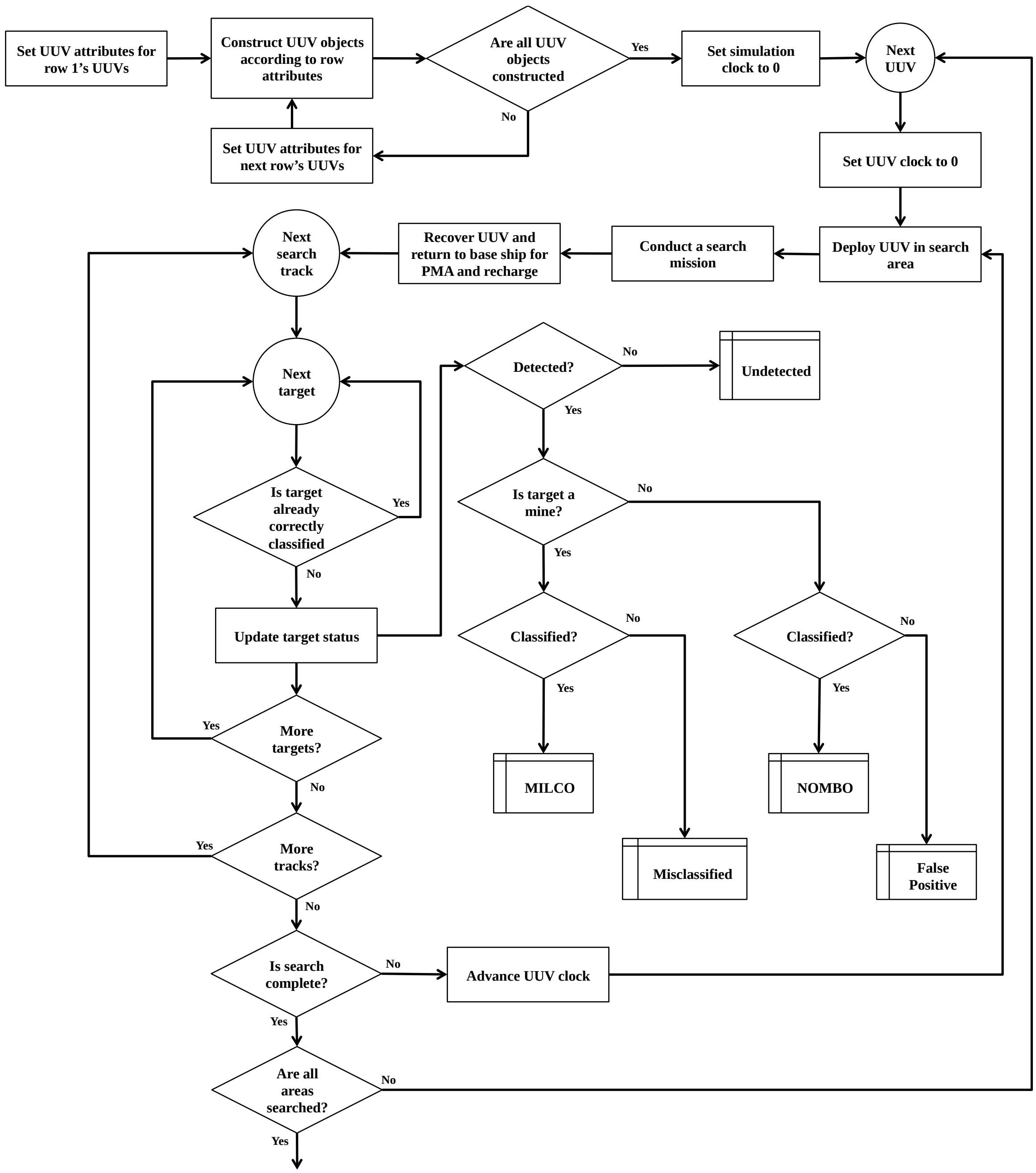



Figure 6. Flowchart showing the logic of the search phase.

### 3. Reacquisition and Identification

The next phase in the scenario is reacquiring and identifying all MILCOs. The UUVs are deployed at the closest MILCO in their areas. The UUV conducts a star pattern inspection and then moves to the next closest MILCO. Each star pattern consists of 20 five-meter tracks. This model uses the standard pattern, not the modified pattern described in Chapter II. The UUV is not capable of inspecting multiple MILCOs in one star pattern. Every MILCO requires its own inspection. PMA is conducted once the UUV is recovered and returned to the ship. This process determines which MILCOs are mines and which are false positives. Figure 7 shows the flowchart for this phase.

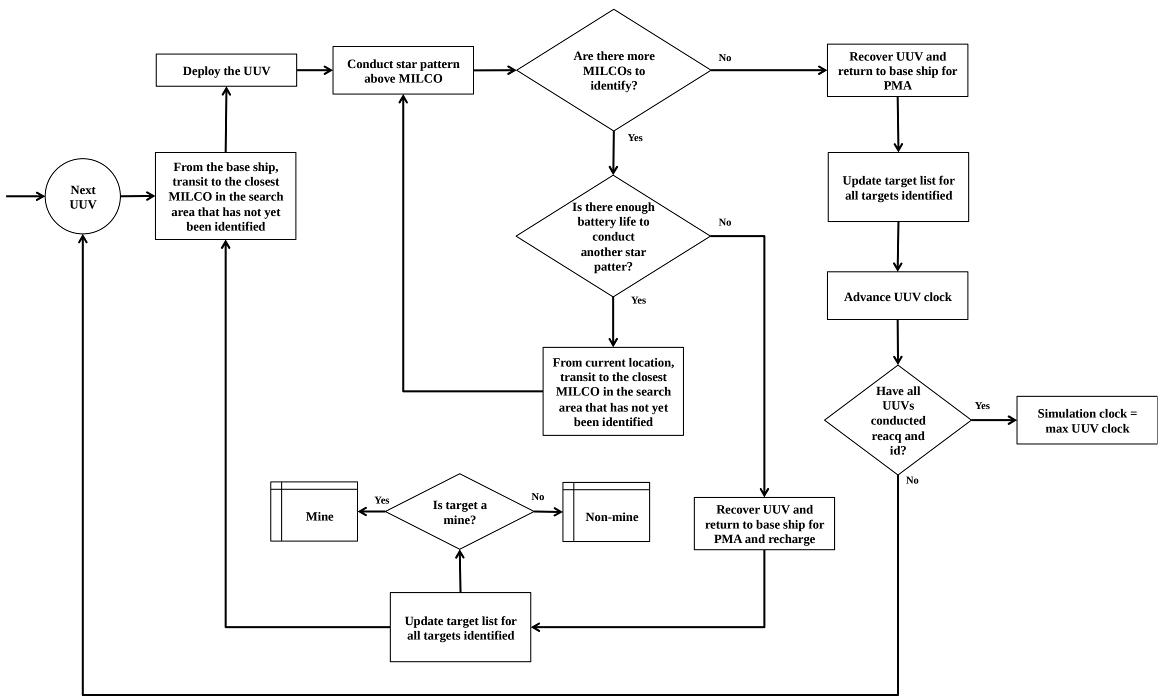



Figure 7. Flowchart showing the logic of the reacquisition and identification phase.

### 4. Neutralization

The final phase of the scenario is the neutralizing all detected mines. Ten EOD dive platoons objects are created and deployed to the Q-route. All EOD dive platoons are paired with the Q-route area object. They do not operate in the smaller search areas, like the UUVs. Instead, they have the ability to operate anywhere in the Q-route. To prevent multiple EOD platoons from operating in the same region, the platoons are assigned a starting position within the Q-route. These starting positions are based on the number of EOD assets and the length of the Q-route. This evenly spreads the EOD platoons across the Q-route. From its starting position, each platoon, in turn, is assigned to neutralize the closest mine that has not yet been neutralized or assigned to another platoon. Once the mine is destroyed, the platoon moves to the next closest mine. The platoon returns to the base ship when it has exhausted its supply of neutralizers or if the maximum time limits are reached. The team rests and resupplies and returns to the Q-route. This process is repeated for all EOD dive platoons until all mines are destroyed. All EOD dive platoons can neutralize mines without incident. Figure 8 shows the flowchart for this phase.

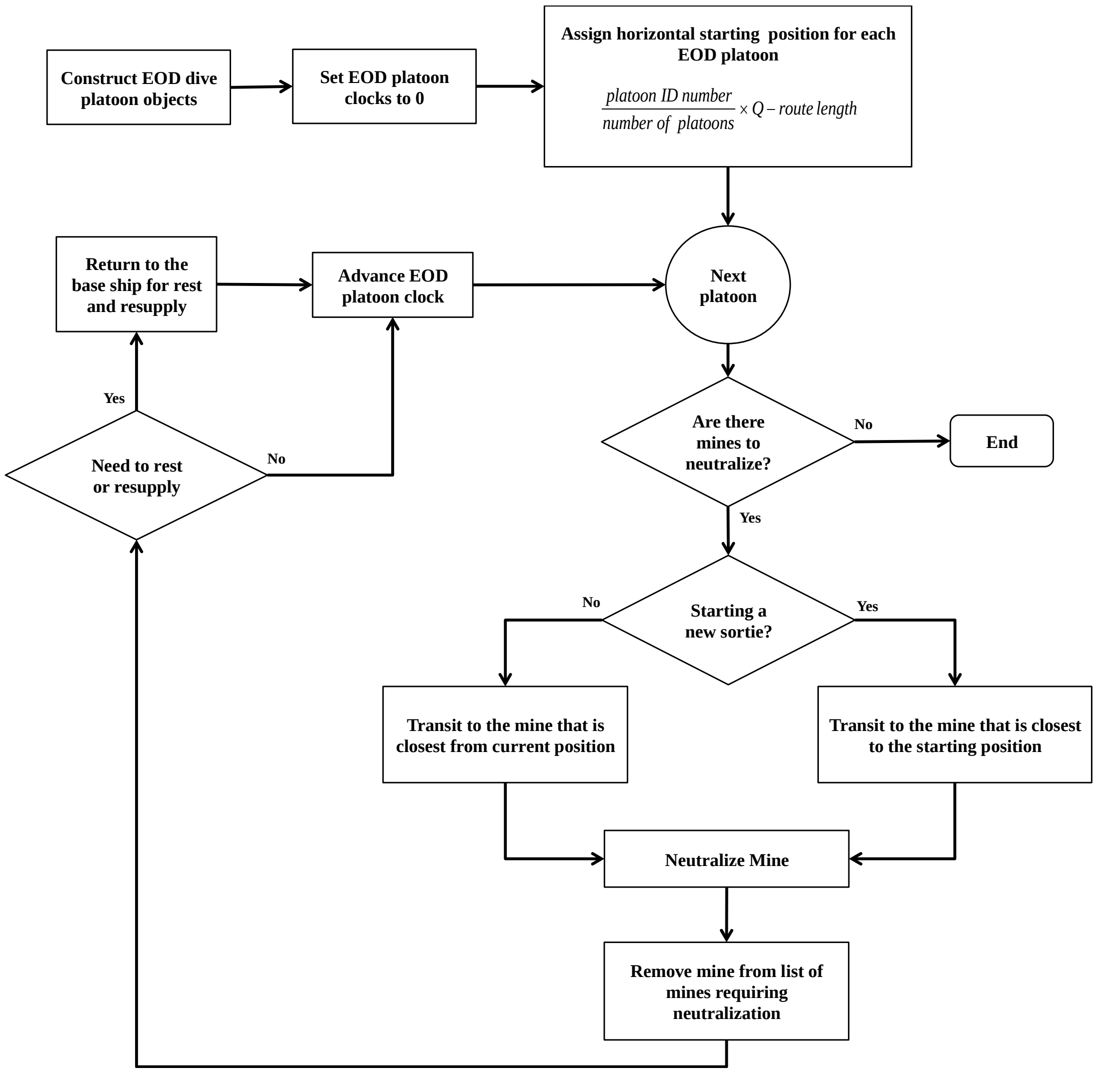



Figure 8. Flowchart showing the logic of the mine neutralization phase.

## C. FACTORS AND RANGES

The simulation has 66 input variables, or factors, that are explored. Four of those variables describe EOD platoon attributes. Two specify the mine density and the clutter density. The remaining sixty characterize the UUVs. While each UUV object requires only twelve inputs, the scenario incorporates five different types of UUVs, one for each row in the Q-route. Each type requires its own set of twelve inputs. The EOD dive platoons require four factors and the Q-route requires two.

These factors are described in Table 1. The high and low levels are approximations made by the author and a subject matter expert. The search area parameters are developed based on the author's previous experience as an MCM warfighter.

Table 1. Factors and ranges used in the simulation experiment.

| Input Variables | Description | Min Values | Max Values |
|---|---|---|---|
| densityMines | Number of mines per square mile | 1 | 10 |
| densityNonMines | Number of non-mines per square mile | 10 | 40 |
| sortieTime | Length of time an EOD dive platoon can remain in the Q-route | 6 hrs | 8 hrs |
| restTime | Length of time an EOD dive platoon must rest before returning to the Q-route | 8 hrs | 10 hrs |
| timeMine | Time spent neutralizing a mine | 1 hr | 2 hrs |
| resupply | Number of neutralizers per sortie | 3 | 6 |
| transitSpeed 1-5 | Speed of the RHIB that is tansporting the UUV | 10 kts | 25 kts |
| deploy 1-5 | Time spent deploying the UUV for a mission | 2 min | 10 min |
| recover 1-5 | Time spent recovering the UUV after a mission | 2 min | 10 min |
| searchSpeed 1-5 | The speed of the UUV during a mission | 3 kts | 5 kts |
| searchTime 1-5 | Length of time the UUV can conduct a mission | 4 hrs | 10 hrs |
| sensor 1-5 | Sensor range of the side scan sonar | 150 meters | 300 meters |
| passes 1-5 | Number of passes per track | 1 | 3 |
| spacing 1-5 | Distance in between search tracks | 30 meters | 100 meters |
| altitude 1-5 | The height above the ocean bottom that the UUV searches | 2 meters | 10 meters |
| detRate 1-5 | The ability of the PMA to detect targets | 0.05 | 0.1 |
| milcoRate 1-5 | The ability of the PMA to correctly classify mines as MILCOs | 0.05 | 0.1 |
| nombosRate 1-5 | The ability of the PMA to correctly classify non-mines as NOMBOs | 0.05 | 0.1 |

## D. ASSUMPTIONS

The model uses nine assumptions in order to reasonably scope the problem. They are numbered and listed below.

1. UUVs in each row share similar search capabilities.

2. All mines are Manta Mines (diameter of 0.98 meters).

3. Weather conditions are perfect. There is no sea state, wind, or current. This assumption is valid because UUVs are not generally deployed in bad weather.

4. The water depth is greater than 40 feet, but shallow enough for divers to safely swim to the bottom.

5. The search area is a rectangular Q-route.

6. The time it takes the UUV to turn around for another search track is negligible, and therefore can be treated as instantaneous within the simulation.

7. EOD dive teams perform perfect neutralization operations. The probability of destroying mines is 1.0.

8. The detection and classification rates are shape parameters for the lateral range curve. They do not represent actual attributes, but they are effective in modeling the ability of a post mission analyst.

9. The scenario is a continuous operation. Assets are able to operate at night.

## E. LIMITATIONS

The model follows the events of a real-life scenario; however, not all aspects are represented in the model. These limitations are numbered and listed below.

1. The person conducting the PMA does not experience fatigue. In reality, post mission analysis is a long and tedious job. It is reasonable to assume the operator's alertness declines over time. This model does not account for such a decrease in alertness.

2. There is a RHIB for every UUV. There is enough room to hold 30 UUVs, but not 30 RHIBs. In reality, there would probably be one RHIB for multiple UUVs. This limitation should not have a substantial affect on MCM mission completion times, because the time spent traveling back and forth with RHIBs is small relative to the total time of the operation.

## F. DESIGN OF EXPERIMENTS

The design of experiments uses the Nearly Orthogonal Nearly Balanced Mixed Design (Vieira, NOB_Mixed_512DP_template_v1.xls Design Spreadsheet.) The high and low values for 66 factors of Table 1 are entered into the NOB_Mixed_512DP_template_v1.xls design spreadsheet; "passes1-5" factors are discrete-valued with three levels, "resupply" factors are discrete-valued with four levels, and the rest are continuous. The columns of inputs are then copied and pasted into a comma separated values (CSV) file. Each design point is then replicated 100 times. Orthogonal design allows each factor to independently contribute to the response variables (Vieira Jr., 2013, pp. 1–4). Nearly orthogonal designs allow a very small amount of correlation in order to achieve better space-filling behavior. This facilitates trade-off analysis.

# IV. ANALYSIS

Clearing a minefield takes a tremendous amount of hard work—and despite the effort, there is no guarantee all mines will be removed. Reducing this risk while maintaining a reasonable timeline is the primary objective for all MCM operations. The intent of this chapter is to examine how to accomplish this objective using only UUVs with different abilities, and to describe the process of analysis used in generating this solution. First, we specify the measures of effectiveness. Using these criteria, we examine the data in order to verify that the software is free of glitches and the output appears reasonable. The next part of the analysis is generating a robust metamodel design. We use this metamodel to predict UUV attributes that increase clearance levels and reduce completion time. Finally, we re-run the simulation at the suggested new configuration and evaluate the performance of the predictions. All graphs and metamodels are generated using JMP Version 11.

## A. MEASURES OF EFFECTIVENESS

MCM plans are designed to achieve a certain clearance ratio within a desired amount of time. Therefore the measures of effectiveness, which best characterize these concerns are: the proportion of objects undetected, the number of misclassified targets, and MCM mission completion time.

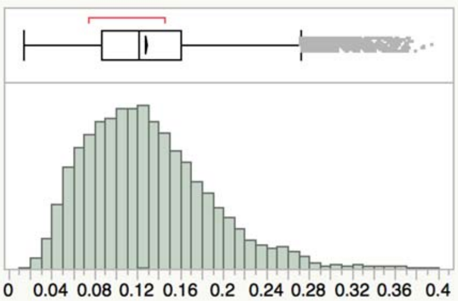
## B. EXAMINING THE DATA

We begin by examining the distribution and summary statistics of the three measures of effectiveness. The histogram shapes and the summary statistics are scrutinized to see if there are any interesting behaviors, in order to verify that the model is behaving properly and that there are no obvious errors.

### 1. Proportions of Undetected Objects

Figure 9 shows the results for the proportion of undetected mines. The distribution has mean of 0.128 and a positive skewness. The proportion of objects ranges from 0.0144 to 0.394. This indicates that every experiment has some undetected objects, and that the proportion is quite high for some cases and low for others. This range variation shows that the model appears to be worth investigating further to determine how much of the changes in output are due to changes in factor levels.

**prop undetected**




**Quantiles**

| | | |
|---|---|---|
| 100.0% | maximum | 0.39444 |
| 99.5% | | 0.31667 |
| 97.5% | | 0.2575 |
| 90.0% | | 0.20222 |
| 75.0% | quartile | 0.16088 |
| 50.0% | median | 0.12148 |
| 25.0% | quartile | 0.08642 |
| 10.0% | | 0.0614 |
| 2.5% | | 0.04304 |
| 0.5% | | 0.0305 |
| 0.0% | minimum | 0.0144 |

**Summary Statistics**

| | |
|---|---|
| Mean | 0.1279082 |
| Std Dev | 0.0557455 |
| Std Err Mean | 0.0002464 |
| Upper 95% Mean | 0.1283911 |
| Lower 95% Mean | 0.1274253 |
| N | 51200 |

Figure 9. The distribution and summary statistics of the proportion of mines undetected.

### 2. Misclassified Targets

Figure 10 shows that no mines are misclassified as NOMBOs during the UUV search. If a mine is detected, it is almost certainly going to be classified correctly. Misclassification appears to be a rare event for all conductions. Similarly, there are no non-mines misclassified as MILCOs during the UUV search. These measures of performance need no further analysis in this thesis, although future experiments could investigate larger ranges of PMA capabilities to see the resulting variation in the primary performance measures. This finding corroborates subject matter expert's previous observations that if objects are detected, misclassification is rare.

misclassified


Quantiles

| | | |
|---|---|---|
| 100.0% | maximum | 0 |
| 99.5% | | 0 |
| 97.5% | | 0 |
| 90.0% | | 0 |
| 75.0% | quartile | 0 |
| 50.0% | median | 0 |
| 25.0% | quartile | 0 |
| 10.0% | | 0 |
| 2.5% | | 0 |
| 0.5% | | 0 |
| 0.0% | minimum | 0 |

Summary Statistics

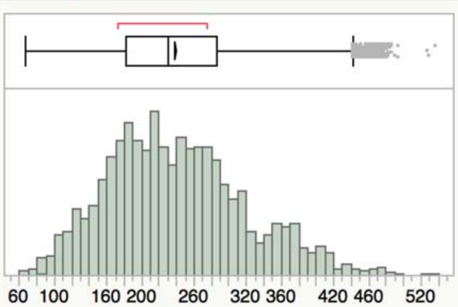
| | |
|---|---|
| Mean | 0 |
| Std Dev | 0 |
| Std Err Mean | 0 |
| Upper 95% Mean | 0 |
| Lower 95% Mean | 0 |
| N | 51200 |

Figure 10. The distribution and summary statistics of the number of misclassified targets.

### 3. MCM Mission Completion Time

Figure 11 shows the distribution and summary statistics for the scenario completion times. The distribution is very wide, indicating a lot of variability. Some of this variability is caused because there are many factors that can extend or reduce the timeline. The mean is 238 hours (9.92 days). The standard deviation is 77 hours (3.21 days).

completionTime




Quantiles

| | | |
|---|---|---|
| 100.0% | maximum | 536.73 |
| 99.5% | | 457.581 |
| 97.5% | | 405.023 |
| 90.0% | | 351.441 |
| 75.0% | quartile | 285.93 |
| 50.0% | median | 230.929 |
| 25.0% | quartile | 181.875 |
| 10.0% | | 144.035 |
| 2.5% | | 105.055 |
| 0.5% | | 82.7636 |
| 0.0% | minimum | 66.8006 |

Summary Statistics

| | |
|---|---|
| Mean | 238.3594 |
| Std Dev | 77.102567 |
| Std Err Mean | 0.3407484 |
| Upper 95% Mean | 239.02727 |
| Lower 95% Mean | 237.69153 |
| N | 51200 |

Figure 11. The distribution and summary statistics of the completion time of the scenario.

## C. ROBUST DESIGN

The proportions of undetected objects and completion times are dependent on two types of attributes: decision factors and noise factors. Decision factors are variables that are controllable by the MCM Commander. These factors are “spacing,” “altitude,” “searchSpeed,” and “resupply.” Noise factors represent variables that cannot be controlled in actual MCM operations. These variables describe the environmental circumstances, such as the number of bottom objects, and the capabilities of UUVs and EOD platoons. A robust design is an analysis technique that identifies ideal decision factor levels that produce acceptable results and are resilient to uncontrollable variation in a system (Sanchez, 2000, p. 70). We use a robust design in this analysis to find ideal factor levels that perform well for two primary measures of effectiveness: the proportion of undetected objects, and MCM mission completion times.

### 1. Summarizing the Data

The model uses a design of experiments with 512 design points and 100 replications. The results are saved into a 51,200 row dataset. This dataset is then condensed into a 512 row dataset by summarizing all 100 replications for each design point into a single row, and calculating the mean and standard deviation for each of the measures of effectiveness. We exclude the noise factors from further analysis because the influence of the noise factors is indirectly captured by the measures. This allows us to focus on the factors that we control in order to find a robust solution.

The distribution of the summarized proportion of undetected objects is shown in Figure 12. The “Mean(prop undetected)” is the distribution of the mean proportion of undetected objects. Its histogram has the same shape as the distribution of proportion of undetected objects in the original dataset. The wide range of outcomes shows the high variation for this response. The average of mean proportions is 0.1279, which is much too high to consider the Q-route safe for transit. The standard deviation of the means is 0.054. This is a high amount of variation. A quarter of the design points produced proportions of undetected objects greater than 0.14. These results are not desirable. The MCM

Commander would not consider the Q-route to be clear without conducting follow-on operations.

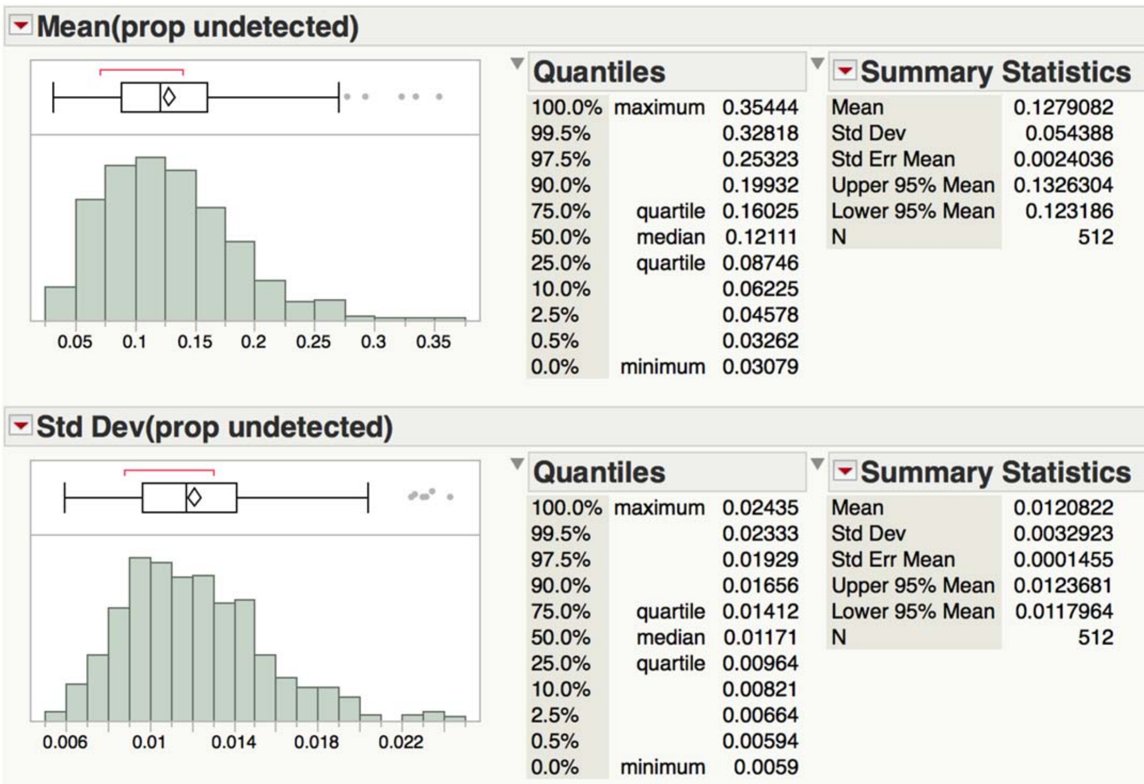



Figure 12. Distribution and summary statistics for the summarized proportions of undetected objects.

The “Std Dev(prop undetected)” shows how much the proportions of undetected objects can changes when all of the variables are the same. These standard deviations range from 0.0059 to 0.0244. This variation is not large, but it is also not small enough to ignore, particularly at the upper end. This means that the performance of the MCM force is not completely predictable. Some operations will yield better results than others, even if the conditions are unchanged.

The distribution of the mean completion times is shown in Figure 13. The mean completion times are extremely varied, ranging from 70 hours (2.92 days) to 474.9 hours (19.79 days). The average completion time is 238.3 (9.93 days) with a standard deviation

of 77 hours (3.21 days). The measure is so varied that it cannot be used to accurately predict the completion time. This is understandable, because the mine density dictates the completion time. There is more work to do in a scenario with more mines. Without intelligence of enemy operations, it is impossible to know *a priori* how many mines are in the water.

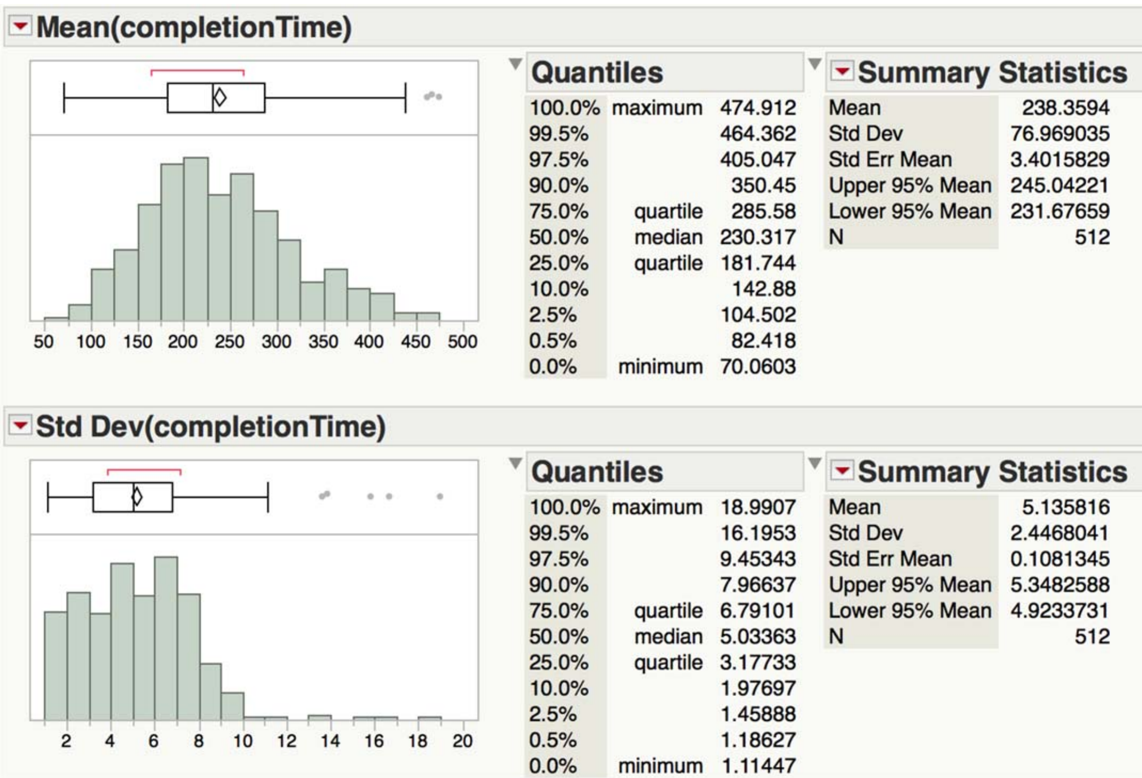



Figure 13. Distribution and summary statistics for the summarized completion times.

The standard deviation of times is much smaller than the variation of the means, but it is still quite large. The maximum standard deviation is about 19 hours, meaning an experiment could have up to 57 hours of variability.

It is assumed in the MCM community that balancing completion times and search efforts is a tradeoff; operations focused on conducting a more thorough search will take a long time, while operations constrained by a quick timeline may leave more objects

undetected. Figure 14 is a scatter plot of average mission completion time versus the average proportion of undetected objects for each design point. The red arrow shows a generally negative relationship between the two measures, as anticipated. The blue oval shows that there are some scenarios where the completion time and the proportion of undetected objects are both low. These experiments show that it is possible to find combinations of decision factors so that both performance measures are close to their respective ideal values.

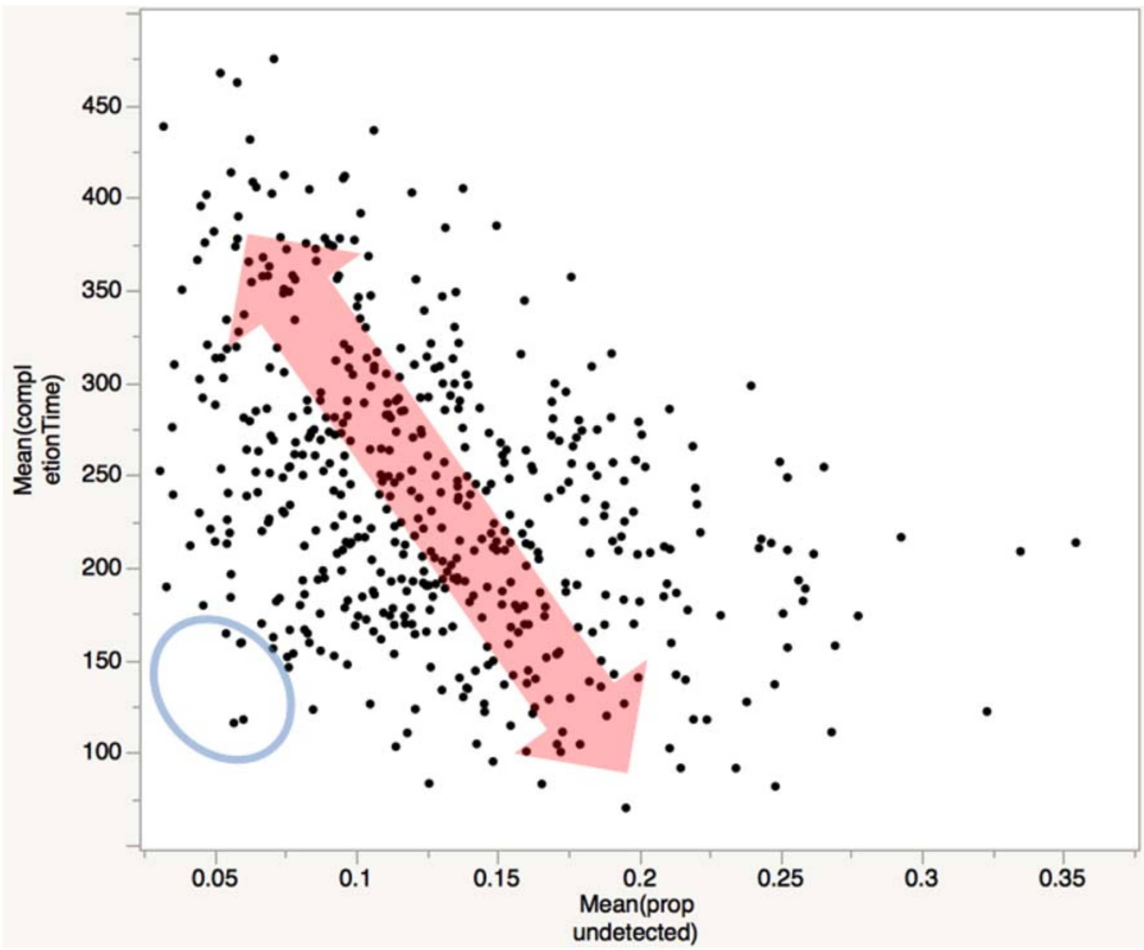



Figure 14. Scatter plot of the mean proportions of undetected objects vs. the mean completion time. The circled design points shows experiments with low proportions of undetected objects and low MCM mission completion times.

### 2. Loss Functions

The condensed dataset summarizes the proportion of undetected objects and completion times into means and standard deviations. The loss function quantifies the performance of the system by assessing the means and standard deviations with reference to a target value. The goal is to achieve an expected solution close to the target output value, while reducing variability of the outcomes. One common loss function is the quadratic, shown below. The equation for this function is written in the following form:

$$l(Y) = (Y - \tau)^2$$
$$E[loss] = \sigma_Y^2 + (Y - \tau)^2$$

where $l(Y)$ is quadratic loss, $E[loss]$ is expected loss, $Y$ is the measure of effectiveness, $\sigma_Y$ is the standard deviation, and $\tau$ is the target value (Sanchez, 2000, p. 71). The target value for the proportion of undetected objects is zero, because the goal is to detect everything. Selecting a target for completion times is not as simple. It makes sense to set the target to zero because it would eliminate the possibility of penalizing completion times that are below the target, but this target does not work well because of the quadratic loss function. The loss of completion times with an accurate range would be too huge to be considered in an analysis. A realistic target for this scenario is one week (168 hours). To prevent penalizing times below one week, we use a modified loss function:

$$expected\ loss(Y) = \begin{cases} 0 & if\ Y < 168\ hrs \\ \sigma_Y + (Y - 168)^2 & otherwise \end{cases}$$

where the loss is 0 if the completion time is less than one week, and quadratic otherwise.

### 3. Metamodels for Expected Loss of Proportion of Undetected Objects

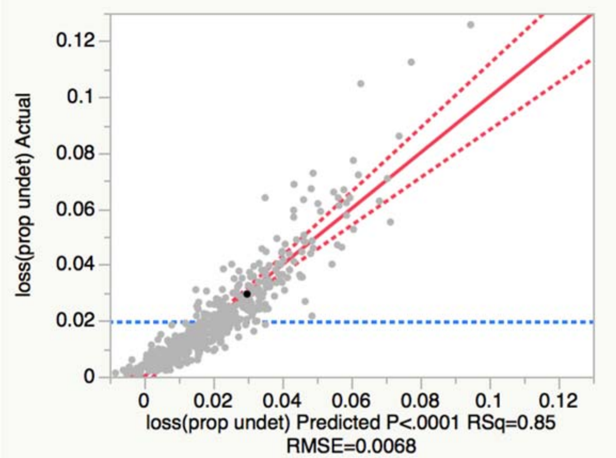
Now that the expected losses are calculated, it is possible to fit metamodels. The general approach includes all main effects, all two-way interaction terms, and all quadratic terms for the decision factors as potential explanatory terms. These factors are

used to conduct a stepwise Bayesian information criterion (BIC) regression to find the appropriate subset of terms that best predict the expected loss. The remaining predictors are then fit to a least squares regression model. Then decision predictors with p-values less than 0.01 are considered significant. We remove predictors that are above 0.01 one at a time, until all predictors are significant. The exception to this rule is if an interaction or quadratic term is significant, but its main effect is not. In that case, we leave the main effect in the model.

Before examining the results and discussing their implications, we recall from Chapter III, Section A, Subsection 1, that the Q-route is divided into five rows. Rows 1 and 5 are on the outside and are the most narrow. Inside rows 1 and 5 are rows 2 and 4. Row 3 consists of the middle, widest row.

We first construct a metamodel of the expected loss associated with the proportion of undetected objects. The resulting regression summary is shown in Figure 15, and the sorted parameter estimates are shown in Figure 16. Figure 16 includes numerals within the factor names to indicate the row number, such as "spacing3" for row 3 or "altitude5" for row 5. Also, the hyphenated numbers following the factor names for interaction and quadratic factors are centering values. For example, "(spacing3-65)*(spacing3-65)" represents the quadratic effect for spacing3, and its average level across all design points is 65 meters. These centering values are for numerical stability purposes, and we will not discuss them further.

## Actual by Predicted Plot




## Summary of Fit

| | |
|---|---|
| RSquare | 0.848563 |
| RSquare Adj | 0.835702 |
| Root Mean Square Error | 0.00682 |
| Mean of Response | 0.01947 |
| Observations (or Sum Wgts) | 512 |

## Analysis of Variance

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 40 | 0.12274258 | 0.003069 | 65.9801 |
| Error | 471 | 0.02190501 | 0.000047 | **Prob > F** |
| C. Total | 511 | 0.14464758 | | <.0001* |

Figure 15. The regression summary for expected loss of proportion of undetected objects.

**Sorted Parameter Estimates**

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| spacing3 | 0.0003625 | 0.000015 | 24.00 | <.0001* |
| altitude3 | -0.002948 | 0.000133 | -22.21 | <.0001* |
| spacing2 | 0.0002008 | 1.524e-5 | 13.18 | <.0001* |
| spacing4 | 0.0001977 | 1.53e-5 | 12.92 | <.0001* |
| altitude2 | -0.001661 | 0.000133 | -12.51 | <.0001* |
| passes3 | -0.004201 | 0.000378 | -11.12 | <.0001* |
| (spacing3-65)*(altitude3-6) | -6.813e-5 | 6.309e-6 | -10.80 | <.0001* |
| altitude4 | -0.001424 | 0.000132 | -10.78 | <.0001* |
| (altitude3-6)*(altitude3-6) | 0.0005982 | 6.636e-5 | 9.02 | <.0001* |
| spacing1 | 0.0001347 | 1.528e-5 | 8.81 | <.0001* |
| passes4 | -0.002655 | 0.000378 | -7.03 | <.0001* |
| altitude1 | -0.000887 | 0.000133 | -6.65 | <.0001* |
| passes2 | -0.002494 | 0.000387 | -6.45 | <.0001* |
| searchSpeed3 | 0.0033488 | 0.000539 | 6.21 | <.0001* |
| (spacing2-65)*(altitude2-6) | -3.352e-5 | 6.479e-6 | -5.17 | <.0001* |
| (altitude4-6)*(altitude3-6) | 0.0002748 | 0.000057 | 4.82 | <.0001* |
| spacing5 | 6.7244e-5 | 1.516e-5 | 4.43 | <.0001* |
| passes1 | -0.001654 | 0.000375 | -4.41 | <.0001* |
| (altitude4-6)*(altitude4-6) | 0.000255 | 6.528e-5 | 3.91 | 0.0001* |
| searchSpeed2 | 0.0020474 | 0.000536 | 3.82 | 0.0001* |
| (spacing3-65)*(spacing3-65) | 3.1863e-6 | 8.73e-7 | 3.65 | 0.0003* |
| (spacing3-65)*(altitude2-6) | -2.451e-5 | 6.733e-6 | -3.64 | 0.0003* |
| altitude5 | -0.000486 | 0.000134 | -3.63 | 0.0003* |
| searchSpeed4 | 0.0018589 | 0.000539 | 3.45 | 0.0006* |
| (altitude2-6)*(altitude2-6) | 0.0002259 | 0.000066 | 3.42 | 0.0007* |
| (altitude3-6)*(spacing2-65) | -2.161e-5 | 6.429e-6 | -3.36 | 0.0008* |
| (passes3-1.93945)*(searchSpeed3-4) | -0.002029 | 0.000644 | -3.15 | 0.0017* |
| (spacing3-65)*(searchSpeed3-4) | 7.9584e-5 | 2.636e-5 | 3.02 | 0.0027* |
| passes5 | -0.00114 | 0.000379 | -3.01 | 0.0028* |
| (passes3-1.93945)*(spacing3-65) | -0.000054 | 0.000019 | -2.85 | 0.0046* |
| (passes2-1.93945)*(spacing2-65) | -5.341e-5 | 1.879e-5 | -2.84 | 0.0047* |
| (altitude3-6)*(altitude2-6) | 0.0001627 | 5.735e-5 | 2.84 | 0.0048* |
| (passes4-2.01953)*(spacing4-65) | -5.247e-5 | 1.855e-5 | -2.83 | 0.0049* |
| (spacing5-65)*(spacing4-65) | 2.1776e-6 | 7.921e-7 | 2.75 | 0.0062* |
| (spacing4-65)*(searchSpeed4-4) | 7.4764e-5 | 2.731e-5 | 2.74 | 0.0064* |
| (passes5-2.04883)*(spacing5-65) | -4.972e-5 | 1.816e-5 | -2.74 | 0.0064* |
| (spacing4-65)*(altitude4-6) | -1.849e-5 | 6.78e-6 | -2.73 | 0.0066* |
| (spacing3-65)*(spacing2-65) | 1.9719e-6 | 7.354e-7 | 2.68 | 0.0076* |
| (searchSpeed2-4)*(altitude1-6) | -0.000608 | 0.000229 | -2.66 | 0.0081* |
| (passes3-1.93945)*(altitude3-6) | 0.0004425 | 0.000167 | 2.66 | 0.0082* |

Figure 16. Sorted parameter estimates for the loss of the proportion of undetected objects.

This model is statistically significant with an F-statistic of 66. The t-statistics and p-values verify that all predictors are significant. The Actual by Predicted Plot in Figure 15 shows our model is not capturing the amount of nonlinearity in the data as well as it might, particularly as the loss increases. This means the model is only partially accurate at predicting the loss, which explains why the $R^2$ is not higher. The $R^2$ value of 0.85 still quite high, and the lack of fit is not a problem within this study because the objective is

not to predict outcomes. The purpose is to identify significant variables in order to create a robust system design.

We would expect the decision factors for row 3 to be more influential because the mines are uniformly distributed throughout the space, which means that row 3 has a disproportionately high object count and factors that strengthen the search in this row should prevail over others. The sorted parameter estimates in Figure 16 show all metamodel factors sorted in order of significance. As expected, the search efforts in row 3 are, indeed, the most influential in determining the loss of proportion of undetected objects. Next are rows 2 and 4, and then rows 1 and 5. Within each row, the most significant decision factor is spacing. As spacing increases, the loss increases. The next most significant factors are UUV altitude and number of passes per track. As these factors increase, the loss decreases. The least influential, but still significant factor, is the search speed. As search speed increases, the loss increases.

Partition trees are also well suited for identifying influential factors; they may be easier to explain to non-technical audiences, and they may do a better job than regression at fitting response surfaces. Figure 17 shows the partition tree for the loss of proportion of undetected objects and Figure 18 shows the leaf report. We use trees to predict responses by starting at the top and following the path down to the leaves based on applicable factor splits. This tree has twenty splits. Each leaf is color-coded. Green leaves represent favorable results, where the mean loss is less than 0.01. Yellow represents mediocre results where the mean loss is between 0.01 and 0.02. Red represents undesirable results where the mean loss is greater than 0.02. The $R^2$ is lower than that of the regression model. This tree is not ideal for prediction, but it does identify influential factors. The most practically important factors are the ones that follow the path to a green leaf. These factors are “altitude3,” “spacing3,” “altitude4,” “altitude2,” “spacing2,” “spacing4,” “altitude1,” “passes3,” and “passes4.” In general, it appears that higher UUV altitudes, smaller track spacing, and more passes, all help reduce the number of undetected objects.

The overall findings are similar to those of the regression model. Row 3 variables are the most influential. The big difference is the order of significant decision factors.

The first split is altitude for row 3 and the second split is spacing for row 3. This differs slightly from the regression model, where spacing for row 3 is most significant and the altitude is the second most significant. Another difference is that row 5 has no splits. This finding does not disagree with the regression model, although the regression showed row 5 factors to be far less important than the other rows. Rows 1 and 5 are the same size and they share similar UUV attributes from the design of experiments. We expect their performance to be comparable.

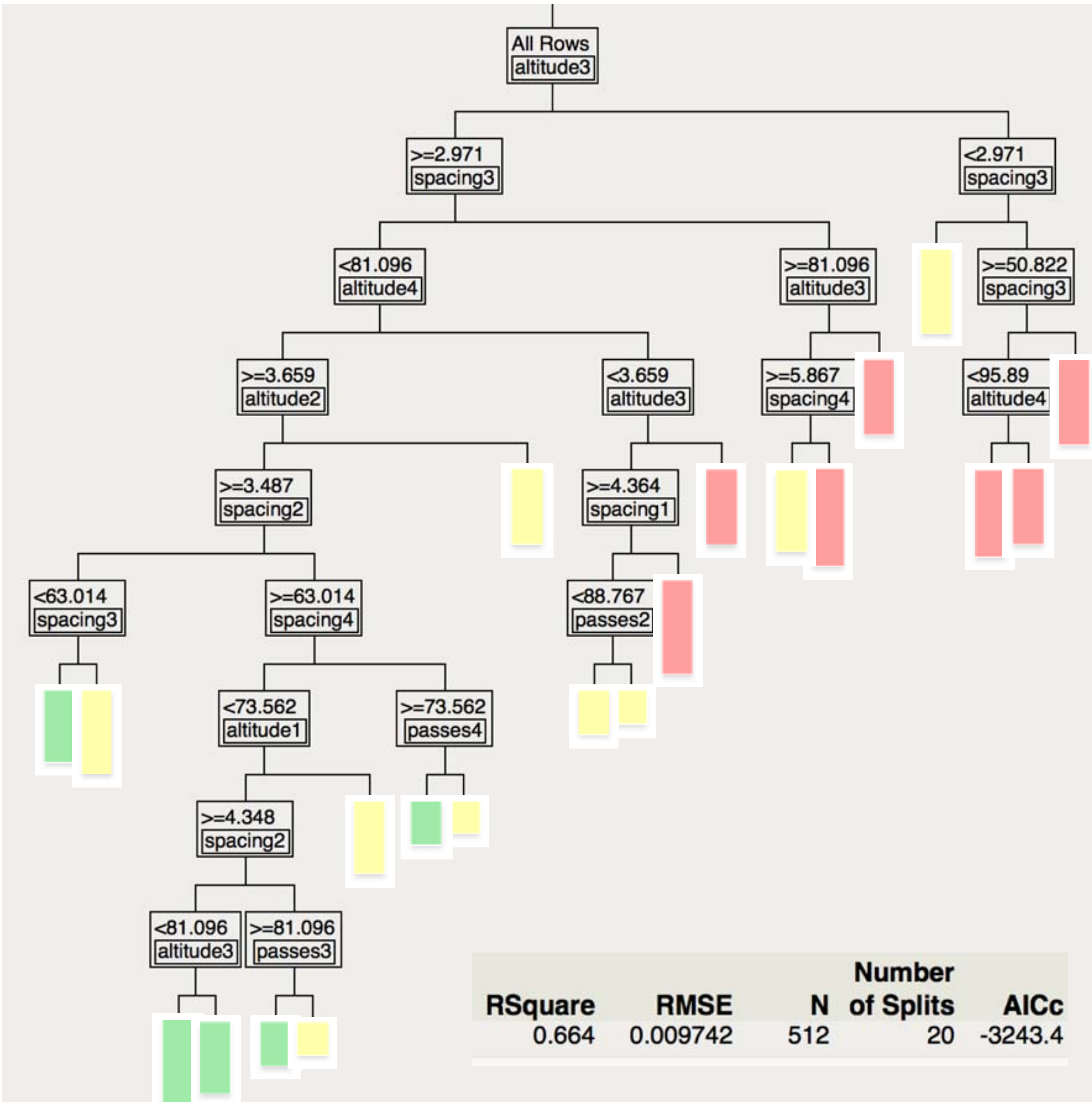



Figure 17. Partition tree model for the expected loss of proportion of undetected objects.

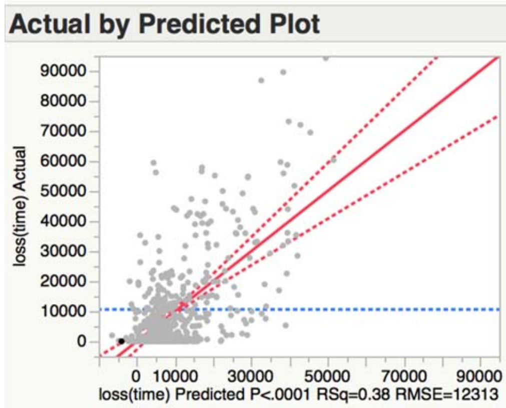
**Leaf Report**

| Leaf Label | Mean | Count |
|---|---|---|
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2<63.014&spacing3<74.11 | 0.00682207 | 99 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2<63.014&spacing3>=74.11 | 0.01387893 | 11 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4<73.562&altitude1>=4.348&spacing2<81.096&altitude3>=5.695 | 0.00431222 | 14 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4<73.562&altitude1>=4.348&spacing2<81.096&altitude3<5.695 | 0.00958392 | 8 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4<73.562&altitude1>=4.348&spacing2>=81.096&passes3>=3 | 0.00719358 | 10 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4<73.562&altitude1>=4.348&spacing2>=81.096&passes3<3 | 0.01298412 | 20 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4<73.562&altitude1<4.348 | 0.01527732 | 13 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4>=73.562&passes4>=3 | 0.00861894 | 10 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2>=3.487&spacing2>=63.014&spacing4>=73.562&passes4<3 | 0.0186261 | 35 |
| altitude3>=2.971&spacing3<81.096&altitude4>=3.659&altitude2<3.487 | 0.0172913 | 45 |
| altitude3>=2.971&spacing3<81.096&altitude4<3.659&altitude3>=4.364&spacing1<88.767&passes2>=3 | 0.00858162 | 16 |
| altitude3>=2.971&spacing3<81.096&altitude4<3.659&altitude3>=4.364&spacing1<88.767&passes2<3 | 0.01606097 | 32 |
| altitude3>=2.971&spacing3<81.096&altitude4<3.659&altitude3>=4.364&spacing1>=88.767 | 0.03587599 | 5 |
| altitude3>=2.971&spacing3<81.096&altitude4<3.659&altitude3<4.364 | 0.03656628 | 14 |
| altitude3>=2.971&spacing3>=81.096&altitude3>=5.867&spacing4<86.164 | 0.01871227 | 63 |
| altitude3>=2.971&spacing3>=81.096&altitude3>=5.867&spacing4>=86.164 | 0.03314237 | 16 |
| altitude3>=2.971&spacing3>=81.096&altitude3<5.867 | 0.03299172 | 39 |
| altitude3<2.971&spacing3<50.822 | 0.01451091 | 11 |
| altitude3<2.971&spacing3>=50.822&spacing3<95.89&altitude4>=3.691 | 0.0388393 | 35 |
| altitude3<2.971&spacing3>=50.822&spacing3<95.89&altitude4<3.691 | 0.06155035 | 11 |
| altitude3<2.971&spacing3>=50.822&spacing3>=95.89 | 0.0844909 | 5 |

Figure 18. Leaf report of the partition tree for the expected loss of proportion of undetected objects; this provides the leaf description, along with the mean proportion of undetected mines and the number of design points associated with each leaf.

### 4. Metamodels for Expected Loss of Completion Times

The analysis of completion times follows the same processes as the analysis for the proportion of undetected objects. We generate metamodels in order to observe trends and identify important decision variables. First, a stepwise BIC regression model is fit with all two-way interaction and second-degree polynomials for the decision factors. The insignificant (or less significant) factors are filtered out, resulting in a parsimonious model with statistically significant factors and interactions. Figure 19 shows the regression summary table and Figure 20 shows the sorted parameter estimates.




**Summary of Fit**

| | |
|---|---|
| RSquare | 0.383402 |
| RSquare Adj | 0.372347 |
| Root Mean Square Error | 12313.38 |
| Mean of Response | 10606 |
| Observations (or Sum Wgts) | 512 |

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 9 | 4.7327e+10 | 5.2586e+9 | 34.6827 |
| Error | 502 | 7.6113e+10 | 151619427 | Prob > F |
| C. Total | 511 | 1.2344e+11 | | <.0001* |

Figure 19. The regression summary for expected loss of MCM mission completion times.

The p-value for the F ratio confirms that the model is statistically significant, but the $R^2$ is extremely low. The Actual by Predicted Plot in Figure 20 illustrates the poor prediction power of this metamodel. The regression line does not appear to follow the data. The plot also shows the high amount of variability in the expected losses because the data are more spread out.

Nonetheless, the regression shows that the decision factors that have the greatest influence over the loss of completion times are track spacing and the number of passes per track. These results follow the same trend as the proportion of undetected objects. The decision factors in Row 3 are the most influential. The main difference in trends is that rows 1 and 5 have no statistically significant terms in the model. This finding is very important. It infers that completion times are not restrained by UUV searches in rows 1 and 5. Less capable UUVs can be tasked to search in these rows. However, if the center row is too large, more UUVs must be assigned to it, as one is not sufficiently capable.

**Sorted Parameter Estimates**

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| spacing3 | -268.1812 | 27.00995 | -9.93 | <.0001* |
| passes3 | 5946.8818 | 674.9269 | 8.81 | <.0001* |
| (passes3-1.93945)*(spacing3-65) | -245.3516 | 33.19158 | -7.39 | <.0001* |
| searchSpeed3 | -4207.123 | 943.7223 | -4.46 | <.0001* |
| (spacing3-65)*(spacing3-65) | 5.6639584 | 1.499006 | 3.78 | 0.0002* |
| (passes3-1.93945)*(searchSpeed3-4) | -4151.052 | 1125.041 | -3.69 | 0.0002* |
| passes2 | 1944.8802 | 679.2643 | 2.86 | 0.0044* |
| spacing4 | -77.10686 | 27.00808 | -2.85 | 0.0045* |
| (passes3-1.93945)*(passes3-1.93945) | 3146.6323 | 1170.871 | 2.69 | 0.0074* |

Figure 20. Sorted parameters estimates for the expected loss of MCM mission completion times.

Next, we create a partition tree for the loss of completion times, and compare its results to those from the corresponding regression metamodel. This tree has 20 splits (Figure 21). The $R^2$ is 0.53, which is considerably higher than the $R^2$ for the regression model, but still not high enough to be an accurate predictor. Again, this is not a bad thing, as the model is not being used to predict outcomes. It is identifying influential factors and attempting to find levels that will reduce loss. This tree follows a similar color scheme as the tree for proportion of undetected objects. Green is used when the mean loss is less than 5,000. Yellow indicates the mean loss is between 5,000 and 10,000. Red shows a mean loss of greater than 10,000.

Many of the findings coincide with those from the regression metamodel for expected loss. Row 3 is the most influential row. In general, the number of passes, track spacing, and search speed are the most influential types of factors. However, there are some interesting differences. The loss metamodels include factors from row 5 in the partition tree; row 5 was strangely insignificant in the analysis of the proportion of undetected objects. These results show that row 5 is a significant contributor to MCM mission completion times.

Currently, the model does not output times for conducting reacquisition and identification. In order to estimate how long this phase takes, we made a rough calculation using several assumptions. A mine density of 10 in the center row produces 15 mines. It should take no longer than 10 minutes to drive from one MILCO to the next

and conduct a star pattern. The UUV completes all star patterns inspections and finishes at the opposite end of the area. It might take just over an hour to travel five NM back to the RHIB. The total time in the water should be no longer than 2.75 hours. The transit time to and from the base ship is one hour. The PMA is then another 2.75 hours. These times add up to 6.5 hours. This is not a long time relative to average times of completion. We can assume that this phase is not creating a bottleneck because it is a relatively fast process.

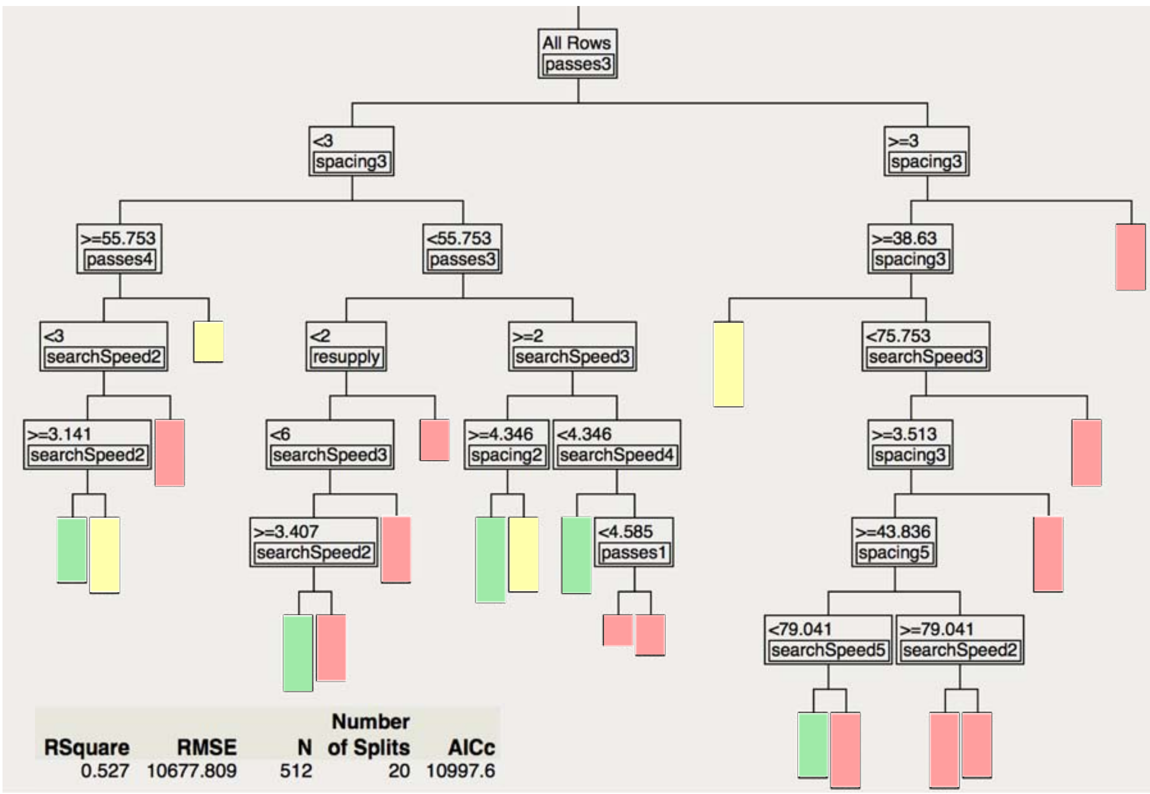



Figure 21. Partition tree for the expected loss of MCM mission completion times.

**Leaf Report**

| Leaf Label | Mean | Count |
|---|---|---|
| passes3<3&spacing3>=55.753&passes4<3&searchSpeed2>=3.141&searchSpeed2<4.953 | 2936.94302 | 136 |
| passes3<3&spacing3>=55.753&passes4<3&searchSpeed2>=3.141&searchSpeed2>=4.953 | 9751.02473 | 6 |
| passes3<3&spacing3>=55.753&passes4<3&searchSpeed2<3.141 | 11307.2432 | 6 |
| passes3<3&spacing3>=55.753&passes4>=3 | 7660.15305 | 75 |
| passes3<3&spacing3<55.753&passes3<2&resupply<6&searchSpeed3>=3.407&searchSpeed2>=3.282 | 2298.28483 | 37 |
| passes3<3&spacing3<55.753&passes3<2&resupply<6&searchSpeed3>=3.407&searchSpeed2<3.282 | 11190.626 | 5 |
| passes3<3&spacing3<55.753&passes3<2&resupply<6&searchSpeed3<3.407 | 10963.3668 | 15 |
| passes3<3&spacing3<55.753&passes3<2&resupply>=6 | 13721.1988 | 20 |
| passes3<3&spacing3<55.753&passes3>=2&searchSpeed3>=4.346&spacing2>=71.781 | 1684.26431 | 10 |
| passes3<3&spacing3<55.753&passes3>=2&searchSpeed3>=4.346&spacing2<71.781 | 9608.85537 | 7 |
| passes3<3&spacing3<55.753&passes3>=2&searchSpeed3<4.346&searchSpeed4>=4.585 | 4607.86718 | 9 |
| passes3<3&spacing3<55.753&passes3>=2&searchSpeed3<4.346&searchSpeed4<4.585&passes1<3 | 19139.6664 | 20 |
| passes3<3&spacing3<55.753&passes3>=2&searchSpeed3<4.346&searchSpeed4<4.585&passes1>=3 | 33423.6216 | 9 |
| passes3>=3&spacing3>=38.63&spacing3>=75.753 | 7272.43061 | 51 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3>=3.513&spacing3>=43.836&spacing5<79.041&searchSpeed5<4.342 | 4790.5306 | 21 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3>=3.513&spacing3>=43.836&spacing5<79.041&searchSpeed5>=4.342 | 13467.6798 | 14 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3>=3.513&spacing3>=43.836&spacing5>=79.041&searchSpeed2>=3.665 | 11076.0841 | 13 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3>=3.513&spacing3>=43.836&spacing5>=79.041&searchSpeed2<3.665 | 27917.6382 | 8 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3>=3.513&spacing3<43.836 | 29644.6491 | 9 |
| passes3>=3&spacing3>=38.63&spacing3<75.753&searchSpeed3<3.513 | 30684.8439 | 20 |
| passes3>=3&spacing3<38.63 | 49193.7514 | 21 |

Figure 22. Leaf report of the partition tree for the expected loss of MCM mission completion times.

### 5. Analyze Odd Behavior

Row 5 UUVs appeared to be less significant than the other rows. This is evident in the detection loss regression metamodel, where row 1 factors showed higher significance than row 5 factors. It can also be seen in the detection loss tree, where row 1 factors were split twice and row 5 factors did not split at all. This occurrence seems especially odd, considering that in the conceptual model, rows 1 and 5 were essentially the same, and the UUV factor ranges for rows 1 and 5 were the same for the experiment. To better examine this result we inspect the bivariate fits for track spacing for rows 1 and 5 against the expected loss of the proportion of undetected objects (Figure 23).

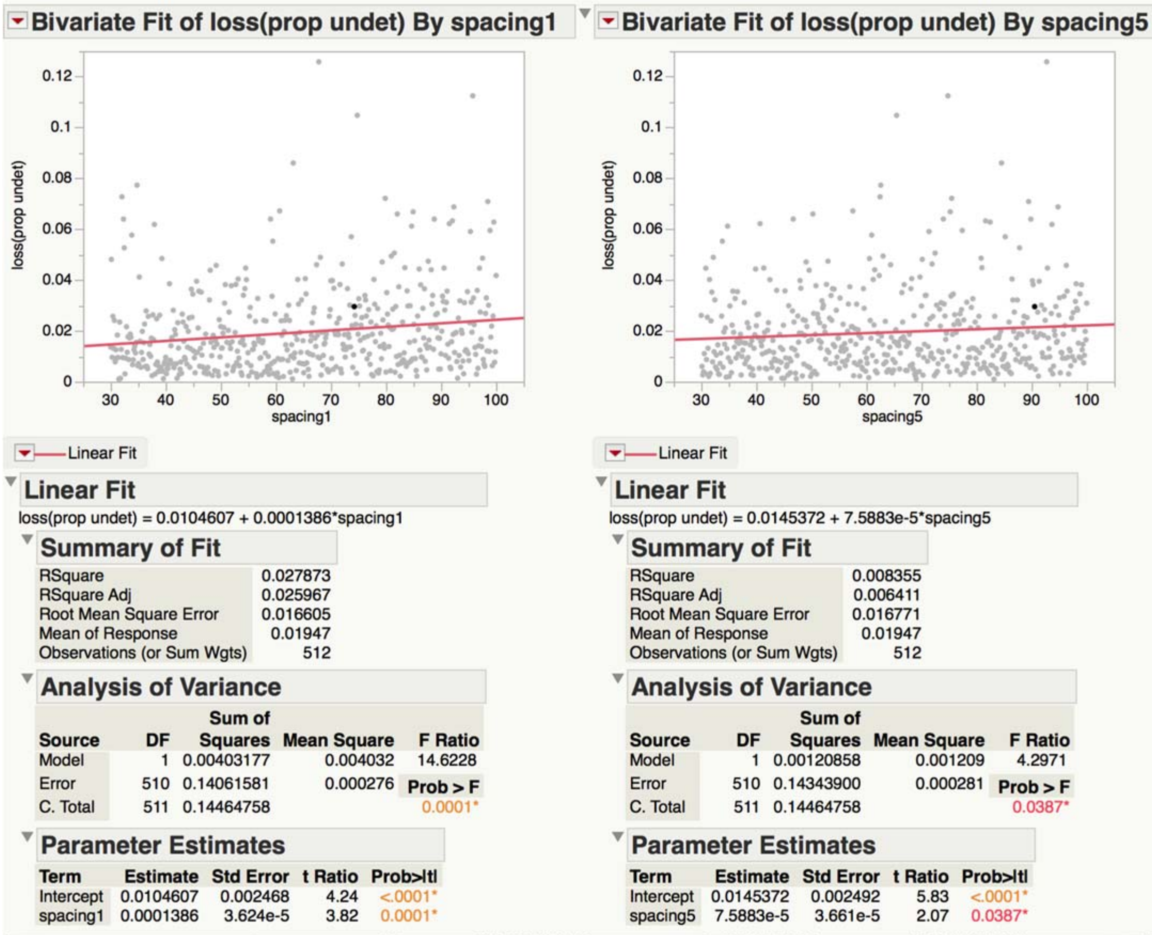



Figure 23. Bivariate fits for track spacing for rows 1 and 5 against the expected loss of the proportion of undetected objects.

The metamodel for row 5 is less significant with an F Ratio of 4.3 and a p-value of 0.039. This shows that row 5 is less important. After running a few exploratory simulations with the model, we determined why this occurs. Figure 24 is presented to explain this behavior. In this simplified example, every search row is one NM wide and thirty NM long. Green circles are neutralized mines. Grey circles are NOMBOs. Red circles are un-neutralized mines and yellow circles are undetected non-mines. The results show the product of UUV search conducted only by the UUVs in rows 1 and 5; UUVs for the inner rows 2–4 are disabled. The reader will notice that the searchers in rows 1 and 5 have secondary contributions to the inner rows based on track geometry. In our model, searches are conducted in the horizontal direction. They start at the bottom left

corner and work their way up. Their last track typically does not line up perfectly with the top border. Therefore, in order to attain full coverage in their own area, they extend one track outside of their areas and into the next row. This extra search track contributes to the search effort in the adjacent row. The amount of extra coverage is dependent on track spacing. Wider tracks allow the UUV to extend further into the adjacent area. This does not happen for row 5 because the last track is conducted outside of the Q-route. Therefore aggregate performance is less sensitive to spacing in row 5, and its contributions are lower than row 1's contributions. The little search effort that row 5 contributes to row 4 happens because the UUV start conducts its first track on the lower border of their search area—and this contribution is independent of track spacing. An operational implication of this is that the anchor point for track demarcation should likely be placed on the inner border of the row in (or close to it) to take advantage of this additional swept area rather than having it overlay ocean floor outside the Q-route south of row 1 or north of row 5.

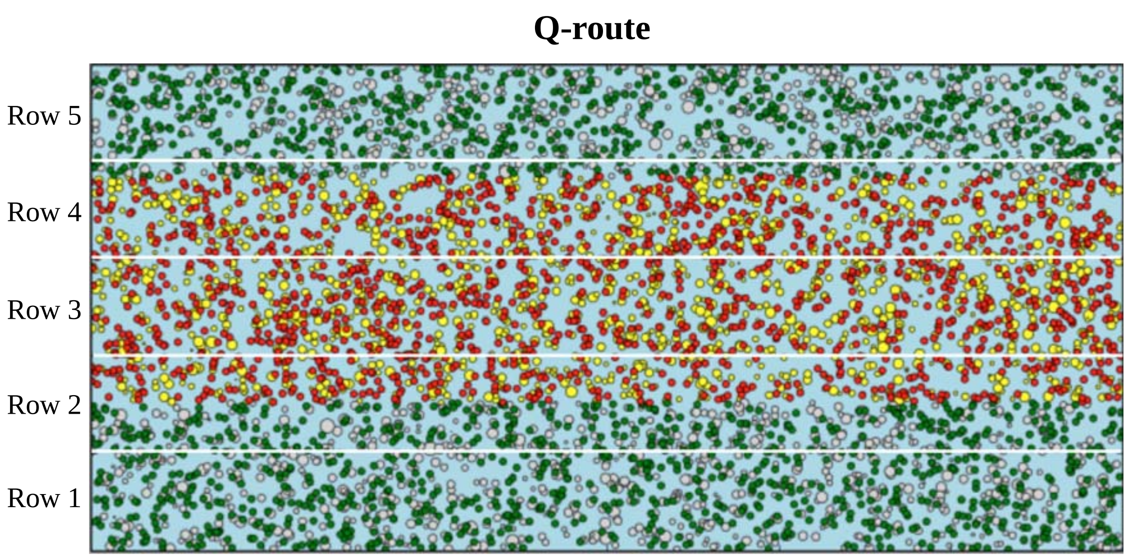



Figure 24. Plot of the Q-route after an exploratory simulation.

This finding demonstrates the important side-benefit of using a large-scaled designed experiment. Behavior that might not have been evident from a single set of

simulation runs (design point) was revealed by the experiment. We were able to track the problem down and use it to refine our model and our insights.

### 6. Significance of Noise Factors

The robust design allows us to center the analysis on the controllable factors. After examining the $R^2$ for the metamodel, it was apparent that some noise factors might be influential. Their absence in the metamodels may have had a considerable effect on the predictive ability of the metamodel. To see how influential these factors might be, we look at partition trees for both measures of effectiveness. To include the noise factors we must revert back to the original output dataset. Figure 25 shows the tree for MCM mission completion times. In ten splits we see mine density splitting twice. This is not a surprise, because the greater number of mines requires more effort from the MCM force. The other noise factor that is split is the time it takes for an EOD platoon to neutralize a mine. This factor is only practically important in high-density minefields. If the MCM Commander has intelligence that the minefield is heavily mined, he or she could request extra EOD platoons to help neutralize all of the mines.

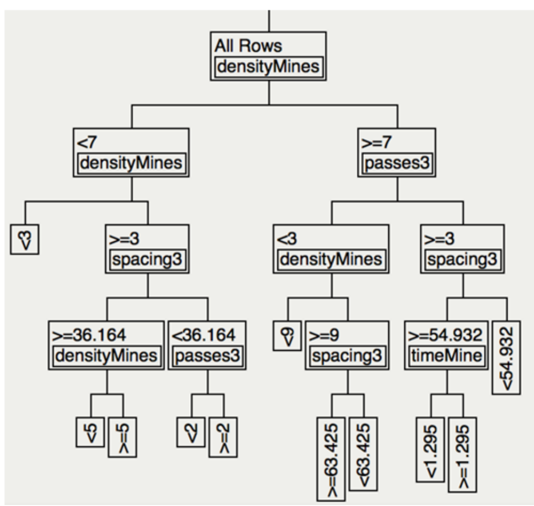



Figure 25. Partition tree for MCM mission completion time with all variables.

Figure 26 shows the partition tree for proportion of undetected objects including all noise factors. This tree is very similar to the tree in Figure 18. There are no noise factors present in the first ten splits. This tells us that we have the ability to control the detection and classification efforts with good planning.

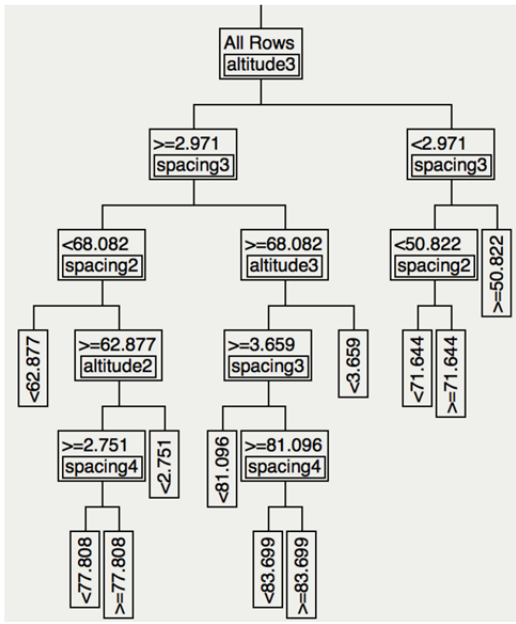



Figure 26. Partition tree for proportion of undetected objects with all variables.

### 7. Baseline Design

In order to evaluate the performance of the ideal design, we must conduct a baseline experiment. We can then compare the two designs and measure the improvement associated with the robust design. The baseline uses factor settings that support fast finish times, but not necessary effective searching. We then separate the UUVs based on capability where detection rates, MILCO classification rates, and NOMBO classification rates are equal for all UUVs per row. UUVs in rows 1 and 5 are poor performers, UUVs in rows 2 and 4 are mediocre performers, and UUVs in row 3 are expert performers. The noise factor levels remain untouched from the original design of experiments. We run this design with ten replications. The results are shown in Figure 27.

The MCM mission completion times are quite low. The MCM commander could finish the operation in 5.19 days, but would not be able to send any ships through, because the Q-route is not clear.

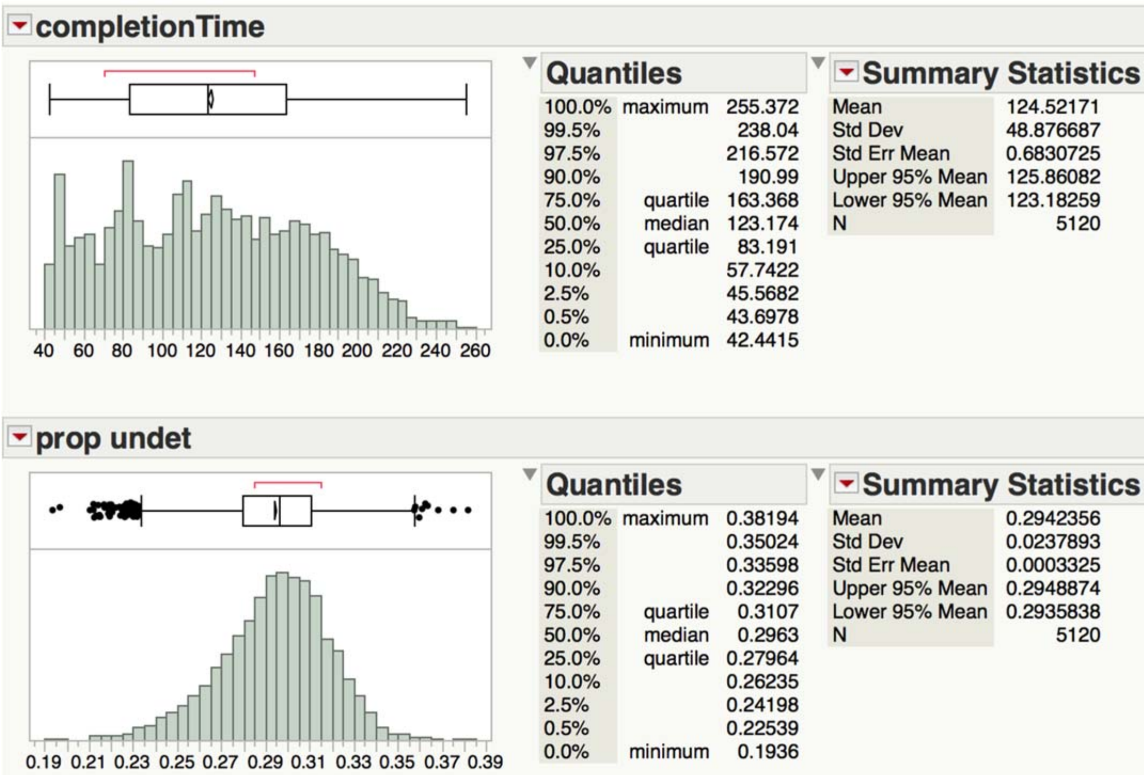



Figure 27. Distributions and summary statistics of the measures of effectiveness for the baseline design.

### 8. Ideal Decision Factor Settings for the Robust Design

The ideal design uses the same scenario as the baseline design, but uses robust decision factor settings. These settings are selected by inspecting the favorable leaf paths described in Figures 19 and 23. Even though row 5 is statistically less significant than row 1, both rows hunt with essentially the same capability. They are equivalent in size and have UUVs with similar abilities; therefore, decision factors selected for one row are implied for the other row. The same process is applied to rows 2 and 4. Table 2 shows the

recommended factor ranges from both partition trees, as well as the selected robust settings.

We decide the final settings based on the UUVs' search capability. UUVs in row 3 are the highly capable, so we assign them settings that support faster mission completion times. Rows 2 and 4 UUVs are mediocre searches, so we assign them mid-level settings. Rows 1 and 5 are the least capable UUVs, and are assigned settings that favor low proportions of undetected objects.

Table 2. Decision factors levels for the robust design.

| Decision Factor | Recommended Levels | Selected Levels |
|---|---|---|
| altitude 1 and 5 | > 4.4 meters | 8 meters |
| spacing 1 and 5 | NA | 45 meters |
| passes 1 and 5 | NA | 3 passes |
| searchSpeed 1 and 5 | NA | 3.5 kts |
| altitude 2 and 4 | > 3.7 | 8 meters |
| spacing 2 and 4 | < 74 meters | 68 meters |
| passes 2 and 4 | < 3 passes | 2 passes |
| searchSpeed 2 and 4 | 3.3 - 5 kts | 4 kts |
| altitude3 | > 3 meters | 8 meters |
| spacing3 | 56 - 74 meters | 73 meters |
| passes3 | 1 - 3 passes | 1 pass |
| searchSpeed3 | > 4.3 kts | 4.3 kts |
| resupply | < 6 neutralizers | 5 neutralizers |

### 9. Results for the Robust Design

The results are displayed in Figure 28. The mean MCM mission completion time for the robust design is 56.6% longer than the baseline design, which is an additional four days. This is not surprising: although the baseline design covered the area faster, the quality of search was so poor that no ships would be able to sail through the Q-route. Therefore, the baseline would require follow-on operations and would ultimately take much longer. Conversely, the robust design was successful at achieving a low mean

proportion (0.06) of undetected objects. The variability was also small (standard deviation = 0.01), indicating that the missions detect objects quite consistently. This is an acceptable clearance level, without resulting in an excessively long mission completion time. The robust design approach worked, by balancing its ability to meet both performance criteria. This experiment shows that with careful planning, UUVs with different abilities can contribute to MCM operations in a variety of minefield conditions.

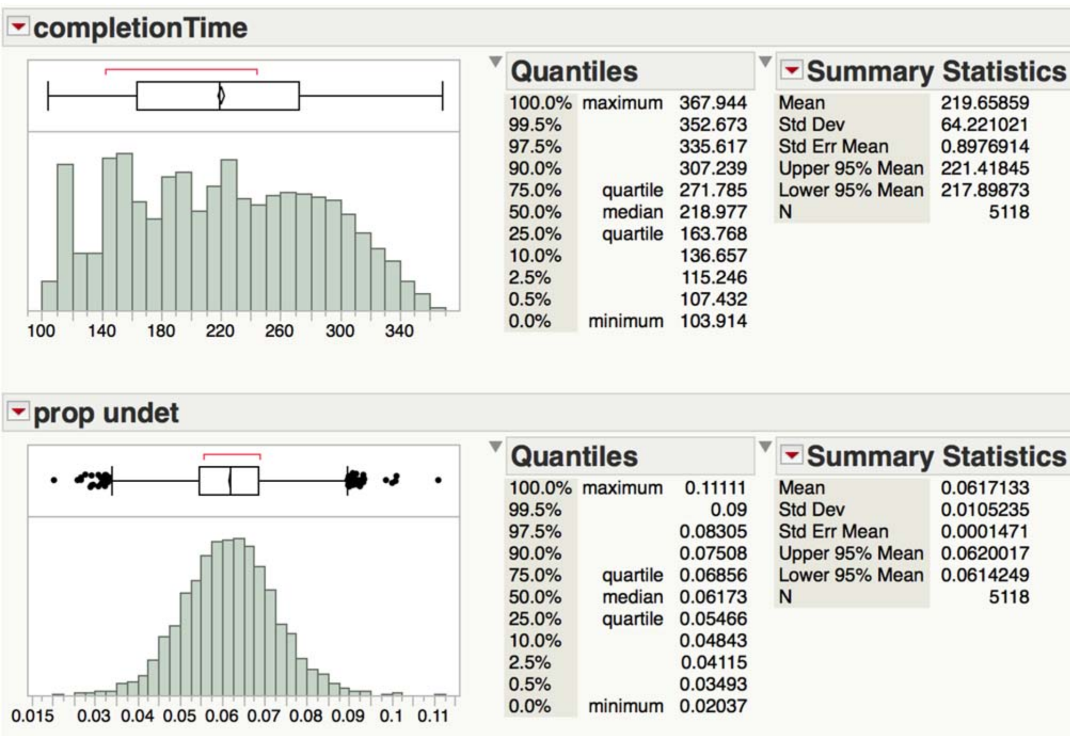
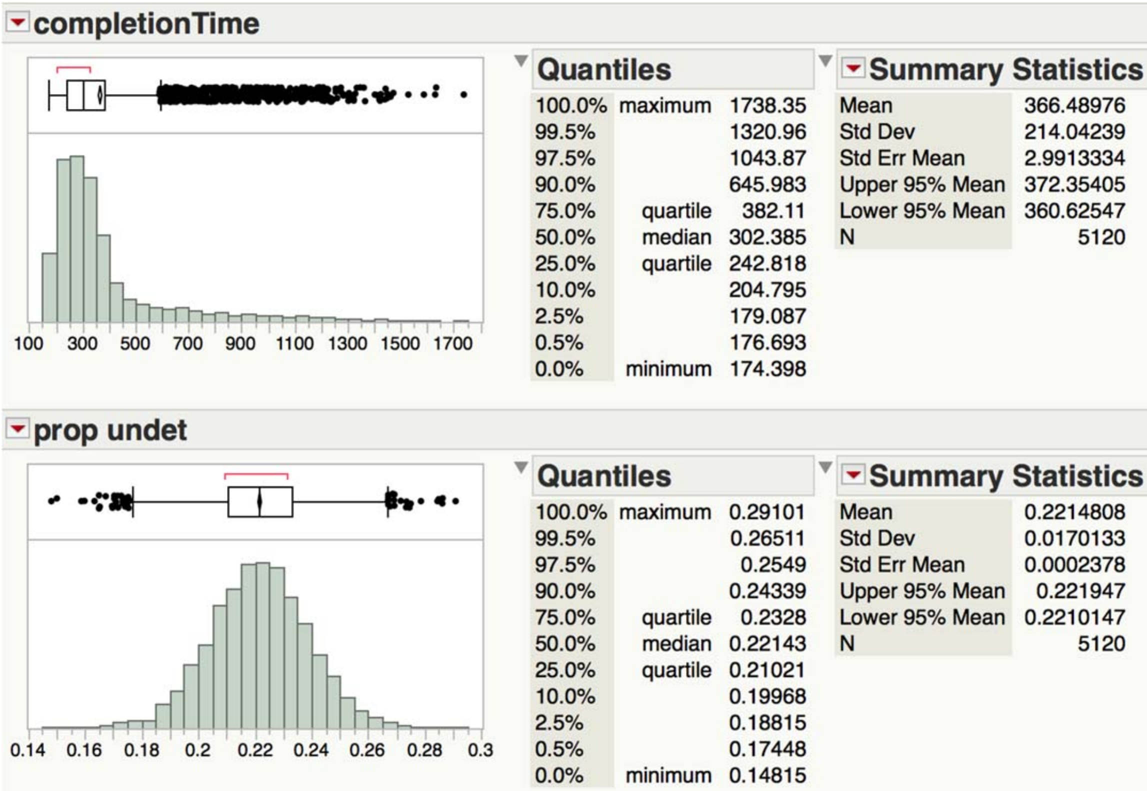



Figure 28. Distributions and summary statistics of the measures of effectiveness for the robust design.

### 10. Results from a U.S.–Only Scenario

For the previous two simulations, we arranged UUVs so that the most experienced ones are in the center row of the Q-route. These UUVs represent U.S.-owned assets. In order to model a scenario with U.S. assets only, we adjusted the code slightly. The dimensions of the Q-route remain the same (30 NM long, 0.9 NM wide), but it is not split into rows. There is only one row composed of six smaller areas and six UUVs. The

rest of the scenario remains the same. The inputs are taken from the new design. The U.S. will use the same ideal settings previously identified. Figure 29 shows the distribution and summary statistics for the MCM mission completion time and proportion of undetected objects. When compared to the robust design, we see that the MCM Mission completion increase dramatically. The mean completion time is 60% higher. Interestingly, the proportion of undetected objects increased by 366%. This is unexpected because the U.S. UUVs have the highest detection rate. This result shows having multiple UUVs hunting side-by-side improves detection. It could also show that because the robust design settings were determined for all UUVs operating together, they are not as effective if conducted individually. Another possibility is that the UUVs in the U.S. scenario did not conduct enough passes. An additional pass could reduce the mean proportion of undetected objects from 0.22 to $(0.22)\cdot(0.22)=0.048$. Therefore in conducting a second search, the search times could be delayed roughly as long as twice the mean, 732 hours.




**completionTime**

| Quantiles | | |
|---|---|---|
| 100.0% | maximum | 1738.35 |
| 99.5% | | 1320.96 |
| 97.5% | | 1043.87 |
| 90.0% | | 645.983 |
| 75.0% | quartile | 382.11 |
| 50.0% | median | 302.385 |
| 25.0% | quartile | 242.818 |
| 10.0% | | 204.795 |
| 2.5% | | 179.087 |
| 0.5% | | 176.693 |
| 0.0% | minimum | 174.398 |

| Summary Statistics | |
|---|---|
| Mean | 366.48976 |
| Std Dev | 214.04239 |
| Std Err Mean | 2.9913334 |
| Upper 95% Mean | 372.35405 |
| Lower 95% Mean | 360.62547 |
| N | 5120 |

**prop undet**

| Quantiles | | |
|---|---|---|
| 100.0% | maximum | 0.29101 |
| 99.5% | | 0.26511 |
| 97.5% | | 0.2549 |
| 90.0% | | 0.24339 |
| 75.0% | quartile | 0.2328 |
| 50.0% | median | 0.22143 |
| 25.0% | quartile | 0.21021 |
| 10.0% | | 0.19968 |
| 2.5% | | 0.18815 |
| 0.5% | | 0.17448 |
| 0.0% | minimum | 0.14815 |

| Summary Statistics | |
|---|---|
| Mean | 0.2214808 |
| Std Dev | 0.0170133 |
| Std Err Mean | 0.0002378 |
| Upper 95% Mean | 0.221947 |
| Lower 95% Mean | 0.2210147 |
| N | 5120 |

Figure 29. Distributions and summary statistics of the measures of effectiveness of the simulation with ideal settings.

## D. DISCUSSION

This analysis demonstrates that combined MCM using UUVs from different countries and with different experience levels can be employed to produces favorable results. The robust design shows us that UUV altitude, track spacing, number of passes per track, and search speed influence the proportion of undetected objects. Since it appears that altitude has no effect on MCM mission completion times, it should always be adjusted to the ideal height. Also, if there is no follow-on mission, then the MCM Commander should increase the number of passes per track, decrease track spacing, and slow search speeds. This will further increase the UUV's abilities to detect bottom objects. This is not always possible. Follow-on missions may require MCM Commanders to conduct MCM operations quickly. This situation highlights the need for fast, but

reliable operations. Analyzing the MCM mission completion times, we found that the decision factors that influence mission completion times are: search speeds, number of passes, and track spacing. By balancing all of the factors, we found that MCM Commanders can conduct effective MCM operations, while minimizing mission completion times. This involves choosing factor settings that satisfy both ideal designs. The results from the test case are evidence that this can be completed. Therefore, MCM Commanders should consider a plan where UUVs are assigned to hunt rows in the Q-route. Less experienced UUVs should remain on the outside of the Q-route and have skinnier search areas. More experienced UUVs should stay on the inside rows and have a wider search area. MCM Commanders are also recommended to consider the decision factors levels from Table 4 when making their MCM plans.

We also examine ability of the U.S. to conduct an MCM scenario alone using UUVs. The results shown in Figure 29 show that the U.S. Navy is not capable of producing efficient results with six UUVs in this scenario. The average completion time increases by 7.7 days, the standard deviation increases by 6 days, and the maximum possible mission completion time increases by two months. While a better MCM plan may improve the results using only six UUVs, the side-by-side comparison is compelling. A combined UUV effort with a range of abilities still outperforms a force of six highly effective UUVs.

Along with the side-by-side comparison of U.S. and coalition performances, we also examine the importance of all factors. The metamodels in the robust design do not include noise factors. This allows us to focus on the factors that we can change. Yet, we learned from Figure 26 that the most influential factor in MCM mission completion times is the density of mines. This figure also shows that high mine densities and low mine densities split on UUV decision factors first. This shows that UUVs are always the greatest cause of delays. The time to neutralize a mine is also significant. EOD neutralizations that take longer than 1.3 hours in mine-dense minefields further delay timelines; however, since neutralization times cannot be controlled by the MCM Commander, the only way to minimize this delay is to recruit more EOD platoons. This

would allow more neutralizations to be conducted simultaneously. These extra assets should reduce the bottleneck caused by high numbers of mines.

Another element that could help MCM operations is the contribution of partner nations. The U.S. Navy is continuously conducting bilateral and multilateral MCM exercises. Therefore, performance of these partners should be evaluated at every opportunity. This information will provide MCM Commanders with valuable information about how and where to employ these partner assets. Exercise evaluators should record mission times, deployment times, recovery times, transit speeds, and general procedures or doctrine that influence they way they conduct their operations. The most important thing to note is their performance. Assigning UUVs areas within the Q-route is important. If they have a low detection rate, then they should be placed in an outer row.

Placing less capable UUVs on the outside of the Q-route does not imply that they contribute less. In fact, the comparison of the robust design and the U.S.-only design suggests the opposite. We expected both minehunting efforts to be comparable. Instead, the robust design, with less capable UUVs, outperformed the U.S.-only scenario. Some of this can be attributable to the total number of UUVs available (30 for the robust design scenario, 6 for the U.S.-only scenario). We can also see the importance of less capable UUVs by inspecting Figure 25. Although a single UUV is assigned to a single search area, its final search area still extends into the adjacent areas, improving the overall detection performance. Figure 25 also reveals that search efforts are being partially wasted in row 5, because the UUVs are searching outside of the Q-route. A more efficient employment tactic would have all searchers start their searches on the outside of the Q-route and work their way inside. This will make use of every search track and improve performance.

## E. MODEL ENHANCEMENTS

This study proves that a combined UUV force can successfully clear a minefield. We have also gained key insight on how to improve operations in different conditions. Though this study is a successful proof of concept, it does not provide MCM

commanders with a tool that they can use to plan operations. Further enhancements need to be made in order to make this model usable. In building this model, we know how to change it in order to better fit the problem. This is not likely a feasible option for the deployed units during operational planning. Therefore this model should be modified to incorporate a large variety of scenarios. It should also make use of a more intuitive input to measure abilities. Probability of detection and classification make sense to most people, more so than a shape parameter. Another necessary improvement would be to automate the process of running a designed experiment, conducting initial follow-on analysis. This is necessary in order to make the program usable in an operational setting.

Another improvement would be to build a graphical user interface (GUI) for the model to make it user-friendlier. This GUI should include an internal data handling process, so that the user would not have to create csv files. Everything would be done internally. This GUI could also include a database of different MCM assets and their attributes, like sensor ranges, mission times, speeds. The user would select the equipment from a list and the GUI would fill in the information automatically. Then after the program has been executed, the GUI will display the answer in an intelligible manor that the MCM Commander can use to make his or her plans. One possibility would be the use of a dashboard, such as that described in that allows trade-offs and feasible alternatives to be visualized graphically.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONCLUSION AND RECOMMENDATIONS

The Avenger-Class fleet will soon retire. The MCM Mission Module progress is at a standstill and the future of MCM is uncertain. Commanders need a practical strategy to ensure the safety of our ships globally. Autonomous vehicles will eventually fulfill this need, but in the meantime, the United States can rely on international partners to help fill this void. A combined UUV force can be used to conduct effective MCM clearance operations in an acceptable timeframe.

## A. SUMMARY

This study used a Python-based simulation model and a design of experiments to generate data. The robust design was performed, and the data was summarized according to a loss function and analyzed using linear regression and partition tree metamodels. Influential decision factors were identified and robust factor levels selected. The design of experiments was modified to reflect these findings, and the scenario was re-run to evaluate the effectiveness of the new mine clearing plan. All measures of effectiveness improved with the new plan. The completion times were reduced by days, and the proportion of undetected objects also drastically declined. This research is evidence that an efficient design can incorporate UUV platoons with a wide range of experience and abilities into a combined force. The outcomes will be effective and efficient MCM operations that can establish the safety of ports and passages worldwide.

## B. RECOMMENDATIONS

This study provides evidence that large-scale MCM operations can be successfully completed using only UUVs. With proper tasking, UUVs with lesser ability levels can be used appropriately and still produce valid results. This study concludes that the following decision factors are influential in conducting clearance efforts: search altitude; track spacing; number of passes. The following decision factors are influential in completion times: track spacing; number of passes per track, search speed, and resupply.

The use of robust decision factor settings could serve as a standard for operational testing and developing formal doctrine for tasking new partner in a large-scale UUV operation. The scenario should follow a similar tasking plan where the Q-route is divided into multiple rows. If the tasking follows this plan and adheres to the operator parameters described in Table 2, then the result will likely be an organized and successful and efficient MCM clearance operation.

## C. FOLLOW ON WORK

The simulation model examines just one scenario. It may be beneficial to explore other scenarios with different types of tasking, new methods, and conceptual platforms. Autonomous UUV systems will soon enter the service. While these systems are safer and faster, there is no precedence established on how to task them. A future project could adapt the code from this project in order to model autonomous capacities and evaluate their performance. This research could give tacticians key insight about how to employ these new systems.

Another follow-on project could examine and compare scenarios where the reacquisition and identification process is eliminated. Figure 10 shows there were no misclassifications during the initial design. If an MCM operation had to be conducted in a short time period, then it may be beneficial to slow the search down to ensure better classification. The likelihood of misclassifying a non-mine as a MILCO would be minimal, as would the expected time loss due to neutralizing false targets is minimal. This might allow MCM Commanders to eliminate the entire reacquisition and identification phase from the operation.

# APPENDIX A. MCM SIMULATION MODEL IN PYTHON

```
import numpy as np
import matplotlib.pyplot as plt


##############################
# General Functions and Variables
##############################

#cartesian calculator
def distCalculator(x1, y1, x2, y2):
    """x1: vector of x coordinates
    y1: array of y coordinates
    x2: array of x coordinates
    y2: array of y coordinates"""
    return np.sqrt((x1-x2)**2 + (y1-y2)**2) #returns a vector of the distance of the two
points

def listStructure(myList):
    #this function takes the array of targets and refits the vectors to their intended
data types
    #x[0], y[1], targetType[2], size[3], detected[4], classified[5], neutralized[6],
actionNeeded[7]
    myList[0].astype(float)
    myList[1].astype(float)
    myList[2].astype(int)
    myList[3].astype(float)
    myList[4].astype(int)
    myList[5].astype(int)
    myList[6].astype(int)
    myList[7].astype(int)
    return myList


##############################
# Search Area Class
##############################

class area(object):
    """Object to build Search Area
    length: length of the area on the x-axis(float type)
    width: width of the area on the y-asis (float type)
    refX: x coordinate of the bottom right corner (float type)
    refY: y coordinate of the bottom right corner (float type)
    encompass: set of all areas that made an area (set)"""

    id = 0 #number of areas created

    def __init__(self, length, width, refX=0, refY=0, encompass=set()):

        self.length = float(length) #length of area in miles (float type)
        self.width = float(width) #width of area in miles (float type)
        self.refX = refX #latitude of bottom left corner
        self.refY = refY #longitude of bottom left corner
        area.id += 1  # increment the counter
        self.id = area.id #id number of the assigned instance
        self.encompass = encompass | set([self.id]) #the set of all encompassing sets


    def __repr__(self):
        #the instance representation
        print self.encompass
        return "len=%.2f, wid=%.2f, position=(%.2f, %.2f), id=%d, encompassing\
```

```
        areas:" % (self.length, self.width, self.refX, self.refY, self.id)


    #places mines and non-mines (assuming only Manta bottom mines)
    def mining(self, densityMines, densityNonMines, targets, sizeMine=0.98, meanSize=1,
stdSize=0.5):
        """densityMines: mines per square mile
        densityNonMines: non-mines per square mile
        sizeMine: diameter of mines (meters)
        meanSize: mean diameter of non-mines (meters)
        stdSize: standard deviation of non-mine diameter (meters)
        targets is array of targets"""

        #Generating the targets and randomly positioning them within the search area
        areaSize = self.length * self.width  #area of search area
        numMines = int(densityMines * areaSize)  #number of mines in the area
        numNonMines = int(densityNonMines * areaSize)  #number of non-mines in the area
        total = numMines + numNonMines
        x = np.random.uniform(0, self.length, total)  #array of random x coordinates for
each target
        y = np.random.uniform(0, self.width, total)  #array of random y coordinates for
each target

        #Determining whether the objects are mines or non-mines
        nonMineType = np.zeros(numNonMines, dtype=bool)  #array of 0s to represent number
of non-mines
        mineType = np.ones(numMines, dtype=bool)  #array of 1s to represent number of
mines
        targetType = np.concatenate((nonMineType, mineType))  #combined array of the 0
array and 1 array
        np.random.shuffle(targetType)  #scrambling the array of mines and non-mines

        #calculating the area of target
        size = np.ones(total) * sizeMine #array where all sizes are set to mine shaped
diameter
        size = np.where(targetType, size, np.random.normal(1,0.3,total))  #logical array
        #if NOMBOS, then reassigns the diameter to a normal random number
        size = np.pi * (size/2.0)**2 #array converting all diameters to areas

        #initializes all shapes to be undetected, unclassified and unneutralized
        detected = np.zeros(total, dtype= bool)  #array of Falses to represent undetected
targets
        classified = np.zeros(total, dtype= bool)  #array of Falses to represent
unclassified targets
        identified = np.zeros(total, dtype= bool)  #array of Falses to represent
unidentified targets
        neutralized = np.zeros(total, dtype=bool) #array of Falses to represent
unneutralized targets

        #populates the scenario mine list into an 8 dimensional array
        newTargets = np.vstack((x, y, targetType, size, detected, classified, identified,
neutralized))

        #concatenates the old targets with the new targets
        targets = np.hstack((targets,newTargets))

        listStructure(targets) #reformating the targets list
        return targets


    #combines areas
    def builder(self, adjoining, dictionary, addLength):
        """adjoining: area to be positioned next to or below the subject area
        dictionary: the areas dictionary
        #addLength: combine "other" left (True) or below (False)"""

        #if the stationary area has no length or width attributes
        if self.length == 0 and not addLength:  #if also adding to the width
```

```
            self.length = dictionary[adjoining].length  #set the equal to the adjoining
area
        if self.width ==0 and addLength:   #if adding to the length
             self.width = dictionary[adjoining].width #set it equal to the adjoining area

        #updating the reference points of the adjoining area
        offsetX = self.refX + self.length * addLength #offset in x direction for the
other area
        offsetY = self.refY + self.width * (not addLength) #offset in y direction for the
other area

        #for each area in the encompassing set of areas in the adjoining area
        for item in dictionary[adjoining].encompass:
            #try because not all encompassing areas exist due to combining areas using
the same name
            try:
                dictionary[item].refX += offsetX #adding to the x position if adding to
the length
                dictionary[item].refY += offsetY #adding to the y position if adding to
the width
            except:
                pass

        #updating the length and width of the new area
        #new length is addition of previous two lengths if adding length
        updateLen = self.length + dictionary[adjoining].length * addLength
        #new width is addition of previous two widths
        updateWid = self.width + dictionary[adjoining].width * (not addLength)

        #union of the sets of encompassing areas for both areas
        encomp = self.encompass | dictionary[adjoining].encompass

        #returning a new area object with new parameters
        return area(length=updateLen, width=updateWid, refX=self.refX,
                    refY=self.refY, encompass = encomp) #returns new area


    #plotting an area
    def plotArea(self, target):
        """This function plots the search area
        light grey:  undetected targets
        blue:          MILCOs
        red:           false negatives
        green:       NOMBOS
        yellow:      false positives
        green:       prosecuted"""


        targets = listStructure(target)

        #x[0], y[1], targetType[2], size[3], detected[4], classified[5], neutralized[6],
actionNeeded[7]
        #arrays
        x = targets[0]
        y = targets[1]
        targetType = targets[2]
        size = targets[3]
        detected = targets[4]
        classified = targets[5]
        neutralized = targets[6]

       #subsetting certain objects in order to colorcode them
        size = size * 20  #setting the size of target markers
        undetected = np.ma.masked_where(detected, size)  #masking everything that has
been detected
        milco = np.ma.masked_where(targetType * detected * classified==False, size)
#masking non-mines and false-negatives
```

```
        falseNeg = np.ma.masked_where(targetType * detected *
np.logical_not(classified)==False, size)  #masking non-mines and MILCOs
        nombos  = np.ma.masked_where(np.logical_not(targetType)  * detected *
classified==False, size)  #masking mines and false-positives
        falsePos  = np.ma.masked_where(np.logical_not(targetType) * detected *
np.logical_not(classified)==False, size)  #masking mines and NOMBOS
        prosecuted = np.ma.masked_where(neutralized==False, size)  #masking everything
that hasn't been prosecuted

        #plotting
        plt.close()  #clear any old plots
        plt.subplot(111, axisbg='lightblue')  #plotting one subplot to make the
background lightblue
        plt.xlim(0, self.length)  #setting x limits
        plt.ylim(0, self.width)  #setting y limits
        plt.scatter(x, y, s=undetected, marker='o', c="lightgrey", linewidth='0.5',
hold='on')  #plotting undetected targets
        plt.scatter(x, y, s=milco, marker='o', c="blue", linewidth='0.5', hold='on')
#plotting MILCOs#
        plt.scatter(x, y, s=falseNeg, marker='o', c="red", linewidth='0.5', hold='on')
#plotting false negs
        plt.scatter(x, y, s=nombos, marker='o', c="green", linewidth='0.5', hold='on')
#plotting NOMBOS
        plt.scatter(x, y, s=falsePos, marker='o', c="yellow", linewidth='0.5', hold='on')
#plotting false pos
        plt.scatter(x, y, s=prosecuted, marker='o', c="green", linewidth='0.5')
#plotting prosecuted targets
        plt.axis('scaled')
        plt.show()

##############################
#
#    UUV Class
#
##############################

class uuv(object):
    """transitSpeed (kts)
        deploy (min)
        recover (min)
        searchTime (hrs)
       searchTime (hrs)
        altitude (meters)
        spacing (meters)
        passes (1, 2, ...)
        sensor (meters)
        setRate (0 - 50)
        milcoRate (0-50)
        nombosRate (0-50)
       originX (-inf, 0)
       originY (-inf, 0)
        """
    id = 0

    def __init__(self, transitSpeed=15, deploy=10, recover=10,
    searchSpeed=4, searchTime=4, altitude=5, spacing=90, passes= 1,
    sensor=3000, detRate=50, milcoRate=50, nombosRate=50, originX=0, originY=0):

        #UUV attributes
        self.transitSpeed = float(transitSpeed)  #transit speed from base to search area
on RHIB
        self.deploy = deploy/60.0  #time to deploy UUV (converted to hrs)
        self.recover = recover/60.0  #time to recover UUV after mission (converted to
hrs)
        self.searchSpeed = float(searchSpeed)  #speed of UUV during search
        self.searchTime = float(searchTime)  #time of missions
        self.altitude = altitude * 0.0005399568  #altitude of UUV during search
(converted to NM)
```

```
        self.spacing = spacing * 0.0005399568  #track spacing (converted to NM)
        self.passes = int(passes)  #number of passes per track
        self.sensor = sensor * 0.0005399568  #track spacing (converted to NM)
        self.detRate = detRate  #detect rate (lateral range curve shape parameter)
        self.milcoRate= milcoRate  #classification rate MILCO (lateral range curve shape
parameter)
        self.nombosRate = nombosRate  #classification rate NOMBOS (lateral range curve
shape parameter)
        self.originX = originX  #staging area x-coordinate
        self.originY = originY  #staging area y-coordinate
        uuv.id += 1  # increment the counter
        self.id = uuv.id

        #working variables
        self.currentTrack = 0  #which track is the UUV searching
        self.currentPass = 0  #current pass for given track
        self.currentX = self.originX
        self.currentY = self.originY
        self.missionClock = 0  #clock per mission
        self.numMissions = 0 #number of missions
        self.clock = 0  #time of completion of last mission
        self.isActive = True #is UUV still searching


    #calculates the probability using inverse square law
    def probability(self, area, targets, ability):

        yCoord = area.refY + self.currentTrack * self.spacing  #determines y coordinate
based on current search track
        y = targets[1] #array of y coordinate of the target
        size = targets[3] #array of the sizes of the target
        cpa = distCalculator(0, yCoord, 0, y)  #array of closest points of approach to
all mines per track
        probability = 1-np.exp(((-2)* ability * size * self.altitude)/
                (self.searchSpeed*(self.altitude**2+cpa**2)))  #array of probabilities
based on inverse cube law
        probability = np.where(cpa < self.sensor, probability, 0) #array setting
probabilities to 0 if out of range
        return probability  #returns array of probabilities

    #conducts search on a track
    def searchTrack(self, area, targets):

        #calculates probabilities to each target for each
        P_d = self.probability(area, targets, self.detRate)  #P{detect}
        P_milco = self.probability(area, targets, self.milcoRate)  * P_d  #P{classify as
MILCO}*Pd
        P_nombos = self.probability(area, targets, self.nombosRate) * P_d #P{classify as
NOMBOS}

        #x[0], y[1], targetType[2], size[3], detected[4], classified[5], identified[6],
neutralized[7]
        #arrays
        targetType = np.array(targets[2])
        detected = targets[4]
        classified = targets[5]
        identified = targets[6]

        #post mission analysis - looking at sonar data
        look = np.random.uniform(0, 1, len(targetType))   #random numbers of each mine
        #take a look if not classified
        targets[4] = np.where(classified, detected, look<P_d) #array
        #classifies MILCO or false negative
        targets[5] = np.where(np.logical_and(targetType==True, classified==False),
look<P_milco, classified) #array
        #classifies NOMBOS or false positive
        targets[5] = np.where(np.logical_and(targetType==False,classified==False),
look<P_nombos, classified)#array
```

```
        #which targets are MILCOS and which are false positives
        #recalculates after search
        detected = targets[4] #array
        #recalculates after search
        classified = targets[5] #array
        #determines if MILCO or false positive
        isMILCO = targetType * detected * classified #array
        isFalsePos = np.logical_not(targetType) * detected * np.logical_not(classified)

        #updates whether identification is needed
        targets[6] = np.where(np.logical_or(isMILCO, isFalsePos),True , identified)
#array

        #increment number of passes per track and adds time to the clock
        self.currentPass += 1
        self. clock += area.length / self.searchSpeed

        return targets

    def mission(self, area, targets):

        #Checks if the mission possible with this UUV
        #UUV must be able to make it down and back in one mission
        possible = (self.searchSpeed * self.searchTime) > (2 * area.length)
        if not possible:
            print "Track too long for this UUV"
            self.isActive = False #finishes up the the UUV's tasking
            return


        #determines how many search tracks are in an area
        totalTracks = int(area.width / self.spacing) + 2 #continues outside of area to
ensure all area is covered

        #Is the mission needed
        self.isActive = (self.currentTrack <= totalTracks) #is the search complete
        if not self.isActive:
            return targets

        #counts number of missions conducted
        self.numMissions += 1


        #UUV mission

        #transit to search area
        yCoord = area.refY + self.currentTrack * self.spacing  #determines y coordinate
based on current search track
        self.clock += distCalculator(self.originX, self.originY, 0, yCoord) /
self.transitSpeed  #transiting to search area

        #deploying UUV
        self.clock += self.deploy  #time to deploy UUV
        tracksThisMission = 0   #current number of tracks searched

        #conduct search
        #assuming the operators recover UUV from same side deployed
        timePerTrack = area.length / self.searchSpeed #time to conduct one search track
        while (self.missionClock + 2 * timePerTrack) < self.searchTime:  #continue if
next 2 tracks don't take too long
                self. missionClock += 2 * timePerTrack  #adding the time of 2 tracks
                tracksThisMission += 2
        self.clock += self.missionClock #adds mission time to the active clock

        #recovering UUV and returning to ship
        self.clock += self.recover  #time to recover UUV
```

```
        yCoord = (tracksThisMission/self.passes) * self.spacing #integer division to 
determine current track
        self.missionClock += distCalculator(self.originX, self.originY, 0, yCoord) / 
self.transitSpeed  #transiting back to ship

        #charging UUV batteries
        self.missionClock = 0 #assumes UUV and team is ready for another mission 
immediately after PMA


        #Post Mission Analysis
        for i in range(tracksThisMission):
            targets = self.searchTrack(area, targets) #doing the PMA for this track
            self.clock += timePerTrack #adding time of PMA for this track

            #after a track is compelte
            if self.currentPass == self.passes:
                self.currentTrack += 1
                self.currentPass = 0  #which track is the UUV searching

        return targets


    def reacquisitionIdentify(self, area, targets):

        #determines if contacts are in the search area
        inRangeX = np.logical_and(targets[0] > area.refX, targets[0] < area.refX + 
area.length)
        inRangeY = np.logical_and(targets[1] > area.refY, targets[1] < area.refY + 
area.width)
        inArea = np.logical_and(inRangeX, inRangeY)

        #time to conduct R&ID "star pattern" with 20 passes at 5 meters per pass
        rID = (20 * 5 * 0.0005399568) / self.searchSpeed #0.0005399568 is the conversion 
from meters to NM

        #finds closest mine for first R&ID mission
        dist = distCalculator(targets[0], targets[1], self.currentX, self.currentY) 
#array of distances to all targets
        dist = np.where(np.logical_and(targets[6], inArea), dist, 10000000)  #distance is 
set to infinity if already identified or out of area
        closest = np.argmin(dist)  #finds the index of the closest mine

        #Reacquisition and identify next target as long as time remains in the mission
        while min(dist) < 10000:  #if distance is less than infinity

            #if first target on mission
            if (self.currentX == self.originX) and (self.currentY == self.originY):
                self.currentX = xRHIB = targets[0][closest] #the UUV and the RHIB is at 
the location of the closest target
                self.currentY = yRHIB = targets[1][closest]
                self.clock += dist[closest] / self.transitSpeed #clock is advanced to 
account for transit
                self.clock += self.deploy #advance the clock for deploying UUV
                self.missionClock += rID #advancing the mission clock for conducting 
first star pattern

                #marks the target as identified
                targets[6][closest] = 0

            else:

                #is there enough time to conduct another R&ID and make it back to RHIB
                distNextTarg = distCalculator(targets[0][closest], targets[1][closest], 
self.currentX, self.currentY) #dist to next target
                distBackToRHIB = distCalculator(targets[0][closest], targets[1][closest], 
xRHIB, yRHIB) #dist from next targ back to RHIB
```

```
                prosecuteTimeNextTarg = (distNextTarg + distBackToRHIB)/self.searchSpeed
+ rID #time to do next R&ID and drive back to RHIB

                #if there is enough time for next R&ID
                if (prosecuteTimeNextTarg + self.missionClock) < self.searchTime:
                    self.currentX = targets[0][closest] #the UUV and the RHIB is at the
location of the closest target
                    self.currentY = targets[1][closest]
                    self.missionClock += distNextTarg/self.searchSpeed + rID #advancing
the mission clock for conducting first star pattern

                    #mark the target as identified
                    targets[6][closest] = 0

                #if there is not enough time, then return to ship to recharge
                else:
                    distToRHIB = distCalculator(self.currentX, self.currentY, xRHIB,
yRHIB) #distance to the RHIB
                    self.clock += distToRHIB/self.searchSpeed #time to transit back to
RHIB
                    self.clock += self.recover #recovery time of the UUV
                    distToShip = distCalculator(xRHIB, yRHIB, self.originX, self.originY)
#dist back to HQ ship
                    self.clock += distToShip / self.transitSpeed #time to transit back to
HQ ship
                    self.currentX = xRHIB = self.originX #update x position once back
onboard the HQ ship
                    self.currentY = yRHIB = self.originY #update y position once back
onboard the HQ ship
                    self.clock += self.missionClock * 2 #advancing clock to account for
the mission plus the post mission analysis/battery charge


            #recalculate distances
            dist = distCalculator(targets[0], targets[1], self.currentX, self.currentY)
#calculates distance to all mines
            dist = np.where(np.logical_and(targets[6], inArea), dist, 10000000)
#distance is set to infinity if already prosecuted or out of area
            closest = np.argmin(dist)  #finds the index of the closest mine

        return targets


##############################
# EOD Dive Team Class
##############################

class diveTeam(object):

    id = 0

    def __init__(self, speed=25, resupply=5, sortieTime=8, restTime=10, originX=0,
originY=0, timeMine=2, timeNonMine=1, isSegment=True, clock=0):
       #EOD Team Attributes
        self.speed = speed #transit speed between mines
        self.speed = float(self.speed) #converts to a float - because not able to
initially assign as float
        self.resupply = resupply  #number of explosives per sortie
        self.sortieTime = float(sortieTime)  #max time allowed per sortie
        self.restTime = float(restTime)  #time between sorties
        self.originX = float(originX)  #staging area x-coordinate
        self.originY = float(originY)  #staging area y-coordinate
        self.timeMine = float(timeMine)  #mean prosecution time of a mine
        self.timeNonMine = float(timeNonMine)  #mean prosecution time of a mine
        self.isSegment = isSegment #is the dive team assigned to a segment of an area
        self.clock = clock #time of last prosecution
        diveTeam.id += 1
        self.id = diveTeam.id
```

```
        #Working variables
        self.currentX = self.originX  #current position x-coord
        self.currentY = self.originY  #current position x-coord
        self.neutOnboard = resupply  #remaining bombs on current sortie
        self.missionClock = 0  #time until next prosecution
        self.isActive = True


    #returns the index of the closest mine shape
    def nearestObject(self, area, targets):

        #x[0], y[1], targetType[2], size[3], detected[4], classified[5], identified[6],
neutralized[7]
        #arrays
        x = targets[0]
        y = targets[1]
        targetType = targets[2]
        classified = targets[5]
        neutralized = targets[7]
        remainingMInes = targetType * classified * np.logical_not(neutralized)


        #determines if the team is on a mission or back at HQ
        isResting = (self.currentX==self.originX)  and (self.currentY==self.originY)

        #gives commander the option dividing the area into segments
        #True: assigns segments based on teams id number
        #       -prevents multiple teams from traveling long distances
        #False: has all teams calculate next closest target based on distance
        #       -near end of scenario, all teams will be going far distances
        #       -longer timeframe
        #       -safer option in case of emergency
        xRef = self.currentX #if on a mission, then the reference point to its current
location

        #if the dive teams are assigned to specific sections
        if self.isSegment:
            if isResting: #if at HQ - sets x reference to the segment
                #the segments are assigned based on number of teams and the dive teams
                xRef = (self.id - 1)* area.length / diveTeam.id

        #arrays of the distances to all objects based on reference point
        dist = distCalculator(x, y, xRef, self.currentY)  #calculates distance to all
mines
        dist = np.where(remainingMInes, dist, 10000000)  #distance is set to remaining
mines - infinity if already prosecuted or not a mine

        #finds the closest mine
        closest = np.argmin(dist)  #finds the index of the closest mine
        if isResting: #recalculates the distance based on current location
            dist = distCalculator(x, y, self.currentX, self.currentY)
        distance = dist[closest]  #captures the distance to the closest mine

        #does dive team have tasking
        if sum(remainingMInes)==0:  #if not then does nothing
            self.isActive = False

        return distance, closest  #returns a tuple to be used in prosecute function


    #function to drive to the next mine and prosecute it
    def prosecute(self, area, targets):

        #transit to next closest target
        nearest = self.nearestObject(area, targets)  #identifies next nearest object
```

```
        #does nothing if no targets to prosecute
        if not self.isActive: #if no MILCOS or false positives, then sit and wait
            return targets

        #if need to return to ship and rest
        isTimeOut = self.missionClock >= self.sortieTime  #is there time left in mission
        noBombs = self.neutOnboard == 0 #are there any bomblets onboard

        if isTimeOut or noBombs: #return to base if time is out or no more bombs
            timeToShip = distCalculator(self.currentX, self.currentY, self.originX,
self.originY)/self.speed
            self.missionClock += timeToShip
            self.currentX = self.originX  #changes location to base
            self.currentY = self.originY
            self.missionClock += self.restTime  #advance the clock to account for rest
time
            self.clock += self.missionClock  #adds the time of mission to the team clock

            #rest and resupply
            self.missionClock = 0 #resets the clock
            self.neutOnboard = self.resupply  #resets number of bomblets



        if not self.isActive: #if no MILCOS or false positives, then sit and wait
            return targets

        distance = nearest[0]  #distance to next closest
        closest = nearest[1]  #index of next closest
        self.missionClock += distance/self.speed  #updates the time taken to transit to
mine

        #conducts prosecution
        underwater = np.random.normal(self.timeMine, 0.5) #adds the time taken to
prosecute a mine (normally distributed with sigma=.5)
        self.missionClock += underwater  #adds to the clock
        self.neutOnboard -= 1 #accounts for the used neutralizer

        #marks the targets as being prosecuted
        targets[7][closest] = 1 #marks the mine as prosecuted

        #update dive team's position
        self.currentX = targets[0][closest]
        self.currentY = targets[1][closest]

        #adds the time of the mission to the clock
        self.clock += self.missionClock

        return targets



##############################
# the scenario
##############################


def secnarioRunner(row):
    """The row should be read in from a csv reader with pre-ordered values"""

    #makes a copy of the input data
    data = list(row) #list
    #pops items from the list to feed into the class instances

    #x, y, target type, size, detected, classified, identified, neutralized
    targets = np.array([[],[],[],[],[],[],[],[]])

    #Dictionary of areas, UUVs and dive teams: keys=id number, values= objects
```

```
areas = {}
uuvs = {}
divers = {}

#resets the class id attribute
area.id = 0
uuv.id = 0
diveTeam.id = 0


#planning process
numUUVs = 30 #number of UUVs available (must be divisible by 5)
numDivers = 10 #number of dive teams
QRouteLength = 30  #length of q-route
rowNames = ["a", "b", "c", "d", "e"] #names of the 5 rows
rowWidths = [.1, .2, .3, .2, .1] #the sizes of the areas

#the HQ ship is just outside of the q-route in safe waters
xHQ = 1 #NM
yHQ = sum(rowWidths)/2.0  #half the distance up on the y-axis

#scenario data
UUVsPerRow = int(1.0 * numUUVs/len(rowWidths)) #UUVs per row in the q-route
areaLen = (1.0 * QRouteLength) / UUVsPerRow #length of each UUV search area

#creates each individual search area
for i in rowWidths:
    for j in range(UUVsPerRow):
        areas[area.id] = area(areaLen, i)

#combining the areas
i = 1
for name in rowNames:
    #builds an empty area for each row
    areas[name] = area(length=0, width=0, encompass=set(name))

    #adds smaller areas to the end of the row area
    for j in range(UUVsPerRow):
        areas[name] = areas[name].builder(i, areas, True)
        i += 1


#creates the combined mine threat area
areas["MTA"] = area(0,0) #creates an empty area for t

#adds rows to the MTA
for name in rowNames:
    areas["MTA"] = areas["MTA"].builder(name, areas, False)


#mining the area
densityNonMines = int(data.pop())
densityMines = int(data.pop())
targets = areas["MTA"].mining(densityMines, densityNonMines, targets)


#building the UUV objects
for i in range(5):
    transitSpeed= float(data.pop())
    deploy = float(data.pop())
    recover = float(data.pop())
    searchSpeed = float(data.pop())
    searchTime = float(data.pop())
    altitude = float(data.pop())
    spacing = float(data.pop())
    passes = int(data.pop())
    sensor = float(data.pop())
    detRate = float(data.pop())
```

```
        milcoRate = float(data.pop())
        nombosRate = float(data.pop())

        #each UUV in a row is built off of the same inputs
        #detRate, milcoRate, and nombosRate are random uniforms numbers +/- .01
        for j in range(UUVsPerRow):
            uuvs[uuv.id] = uuv(transitSpeed=transitSpeed, deploy=deploy,
                                recover=recover, searchSpeed=searchSpeed,
                                searchTime=searchTime, altitude=altitude,
                                spacing=spacing, passes=passes, sensor=sensor,
                                detRate=np.random.uniform(detRate-.01,detRate+.01),
                                milcoRate=np.random.uniform(milcoRate-.01,milcoRate+.01),
                                nombosRate=np.random.uniform(nombosRate-.01,
nombosRate+.01),
                                originX=xHQ, originY=yHQ)

    #initializes the clock to 0
    completionTime = 0

    #UUVs search their entire areas
    for UUV in uuvs:

        #detect, classify and localize
        while uuvs[UUV].isActive:
            targets = uuvs[UUV].mission(areas[UUV],targets)

        #reaquire and identify
        targets = uuvs[UUV].reacquisitionIdentify(areas[UUV],targets)

        #waits until all UUV searches and identifications are complete before starting
neutr
        if uuvs[UUV].clock > completionTime:
            completionTime = uuvs[UUV].clock #the longest search sets the clock


    #Making the dive team objects
    resupply = int(data.pop())
    timeNonMine = float(data.pop())
    timeMine = float(data.pop())
    restTime = float(data.pop())
    sortieTime = float(data.pop())

    #builds the dive team object
    for i in range(numDivers):
        #time for all teams is the completion time of the last UUV search
        divers[diveTeam.id] = diveTeam(timeNonMine=timeNonMine,
                resupply=resupply, timeMine=timeMine,
                restTime=restTime, sortieTime=sortieTime, originX=xHQ,
                originY=yHQ, clock=completionTime)


    #dive teams conduct prosecution until no MILCOs and false positives are left
    while divers[numDivers].isActive:
        #each team conducts 1 prosecution before looping back through
        for team in divers:
            targets = divers[team].prosecute(areas["MTA"],targets)

            #last mine neutralized sets the clock
            if divers[team].clock > completionTime:
                completionTime = divers[team].clock

    #calculates output statistics
    totalTargets = len(targets[1])
    numMines = sum(targets[2])
    numNonMines = sum(np.logical_not(targets[2]))
    numUndetected = sum(np.logical_not(targets[4]))
    numDetected = sum(targets[4])
    numClassified = sum(targets[4] * targets[5])
```

```
numMILCOS = sum(targets[2]*targets[4]*targets[5])
numNOMBOS = sum(np.logical_not(targets[2])*targets[4]*targets[5])
numNotClassified = sum(targets[4] * np.logical_not(targets[5]))
numFalseNeg = sum(targets[2] * targets[4]*np.logical_not(targets[5]))
numFalsePos = sum(np.logical_not(targets[2]) * targets[4]*np.logical_not(targets[5]))

return row + [totalTargets, numMines, numNonMines, numUndetected,
              numDetected, numClassified, numMILCOS, numNOMBOS,
              numNotClassified, numFalseNeg, numFalsePos, completionTime]
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B. PYTHON SCRIPT USED TO RUN THE DESIGN OF EXPERIMENTS

```
import sys
import csv
from UUV_Simulation import *

#experiment design
document = 'test.csv'  #name of DOE file
replications = 100  #number of replications per experiment

#opening an outfile
out_file = open("outters.csv", 'wb') #opening an output write file
owriter = csv.writer(out_file, delimiter=',') #creating a csv writer object

#Parsing the DOE data
in_file = open(document, 'rU')  #opening the file
in_reader = csv.reader(in_file)  #creating a csv reader object

#copying the headers and printing them to the outfile
headers = in_reader.next()
headers = headers + ["totalTargets", "numMines", "numNonMines", "numUndetected",
                     "numDetected", "numClassified", "numMILCOS", "numNOMBOS",
                     "numNotClassified", "numFalseNeg", "numFalsePos", "completionTime"]
owriter.writerow(headers) #writing the headers plus the names of the other

#parsing the data
for row in in_reader:  #examining each row or disaster from the entire data set
    for i in range(replications):  #replicating each experiment
        temp = secnarioRunner(row) #running the scenario
        owriter.writerow(temp) #writing the data to the outfile
in_file.close()
out_file.close()
```

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


Allen, T. E. (2004). *Using discrete event simulation to assess obstacle location accuracy in the Remus Unmanned Underwater Vehicle* (Master's thesis). Retrieved from Calhoun http://calhoun.nps.edu/bitstream/handle/10945/35337/120.pdf?sequence=1

Allen, T. E., Buss, A. H., & Sanchez, S. M. (2004). Assessing obstacle location accuracy in the Remus Unmanned Underwater Vehicle. *Winter Simulation Conference* (pp. 940–948). Washington, DC: INFORMS.

Ashpari, M. J. (2012). *A Capability-based approach to analyzing the effectiveness and robustness of an offshore patrol vessel in the search and rescue mission. A capability-based approach to analyzing the effectiveness* (Master's thesis). Retrieved from Calhoun https://calhoun.nps.edu/bitstream/handle/10945/27785/12Dec_Ashpari_Mohammed.pdf?sequence=1

Bays, M. J. (2014, May 12). Recent advances in mine countermeasure and mine warfare autonomy at NSWC PCD. Retrieved Feb 24, 2015, from www.nps.edu: http://www.nps.edu/Academics/Centers/CRUSER/Events/individualEvents/individualEvents_docs/2014_05_Bays.pdf

Blandin, M., Brux, J., Caraway, C., Cook, J., Fromille, S., Haertel, D., et al. (2014). Software design document models of legacy and future mine countermeaures systems. Naval Postgraduate School, Systems Engineering. Monterey, CA: Naval Postgraduate School.

Chung, T. (2014, April 28). OA3602 Search theory and detection: Lateral range curves. Naval Postgraduate School, Operations Research. Monterey, CA.

Federation of American Scientists. (1999, January 24). MCM-1 Avenger Mine Countermeasures Ships. Retrieved February 12, 2015, from www.fas.org: http://www.fas.org/man/dod-101/sys/ship/mcm-1.htm

Kemp, M., Richardson, B., Love, L., Rodgers, T., Nelson, K., & Tracy, D. (2011). Analysis of manipulator arm steadiness during ship-hull mine neutralization by a UUV. *OCEANS 2011* (pp. 1-5). Waikoloa, HI: IEEE.

Kim, C. (2002). *The effect of sensor performance on safe minefield transit*. (Master's thesis). Retrieved from Calhoun http://calhoun.nps.edu/bitstream/handle/10945/3704/02Dec_Kim_Chihoon.pdf?sequence=1

Leier, S., Fandos, R., & Zoubir, A. M. (2015). Motion error influence on segmentation and classification performance in SAS-based automatic mine countermeasures. *IEEE Journal of Oceanic Engineering, 40*(1), 57-70.

Maurelli, F., Patron, P., & Cartwright, J. (2011). *Integrated MCM missions using heterogeneous fleets of AUVs.* Heriot-Watt University, Ocean Systems Laboratory. Edinburgh: IEEE.

Midnight, B. (2014, November 4). Mine warfare oceanography center gets permanent deployment to 5th Fleet. Retrieved March 23, 2015, from www.navy.mil: http://www.navy.mil/submit/display.asp?story_id=84665

Naval Mine and Anti-Submarine Warfare Command. (2013). *NMAWC table top war game report*. Newport, RI: US Naval War College.

Allen, G. (2013, October 31). *18th NDIA Expeditionary Operations Conference*. OPNAV N95. Portsmouth, VA.

Sanchez, S. M. (2000). Robust design: Seeking the best of all possible worlds. 2000 *Winter Simulation Conference* (pp. 69-76). Monterey, CA: INFORMS Simulation Society.

Seligman, L. (2015, January 26). Challenges persist for LCS Remote Minehunting System in recent tests . *Inside the Navy, 28*(3).

Seligman, L. (2014, 12 22). Cuts to LCS mission modules program will increase costs, delay fielding . *Inside the Navy, 27*(51).

Song, K. s., & Chu, P. C. (2014, February 14). Conceptual design of future undersea unmanned vehicle (UUV) system for mine disposal. *IEEE Systems Journal , 8* (1), pp. 43-44.

Sternlicht, D. D., Fernandez, J. E., & Marston, T. M. (2013, April-June). *Advances in synthetic aperture sonar transform mine countermeasures and undersea warfare.* Retrieved March 17, 2015, from http://www.doncio.navy.mil: http://www.doncio.navy.mil/chips/ArticleDetails.aspx?ID=4535

Sulzberger, G., Manley, R. J., Vaizer, L., Holtzapple , R., & Bono, J. (2009). *Hunting sea mines with UUV-based magnetic and electro-optic sensors.* Panama City, FL: Naval Surface Warfare Center Panama City .

U.S. Joint Chiefs of Staff. (2011). *Joint publication 3-15*. Washington, DC: U.S. Joint Chiefs of Staff.

United Nations. (2015). United Nations convention on the law of the sea. Retrieved January 04, 2015, from www.un.org: //www.un.org/depts/los/convention_agreements/texts/unclos/unclos_e.pdf

Vieira Jr., H. S. (2013). Efficient, nearly orthogonal-and-balanced, mixed designs: An effective way to conduct trade-off analyses via simulation. *Journal of Simulation, 7*, pp. 264-275.

Vieira, J. H. (n.d.). NOB_Mixed_512DP_template_v1.xls Design Spreadsheet. Retrieved January 09, 2015, from http://harvest.nps.edu.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California